Text2Mesh:
Text-Driven Neural Stylization for Meshes

Figure 1. Text2Mesh produces color and geometric details over a variety of source meshes, driven by a target text prompt. Our stylization results coherently blend unique and ostensibly unrelated combinations of text, capturing both global semantics and part-aware attributes.

Project ID: 14

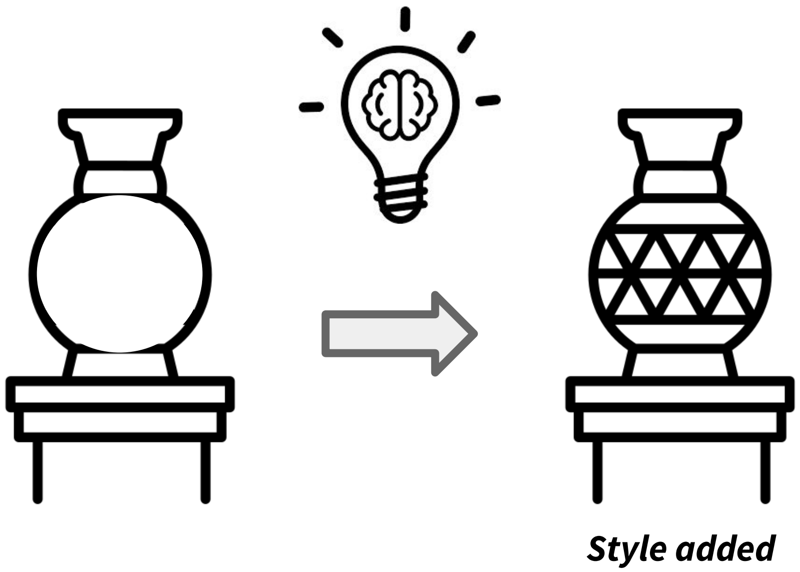
Mooyeol Oh & Muhammad Izaaz Inhar Ramahdani

Development Track

[Michel et al., arXiv 2021]

Motivation

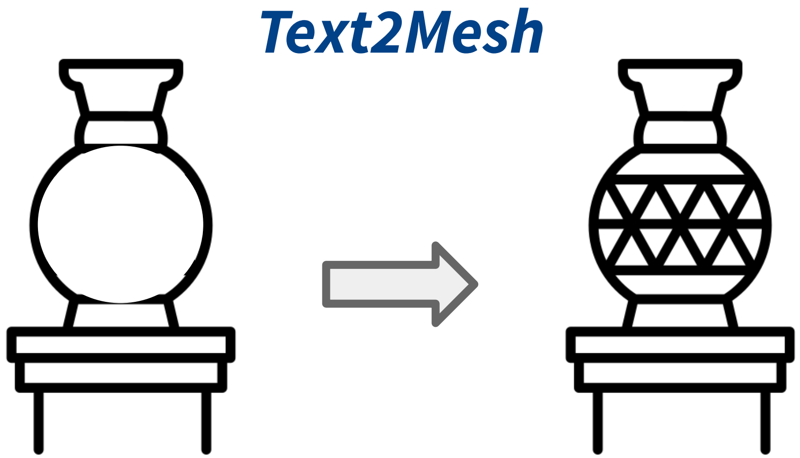
Intuitive control the 3D object by using natural language



- ***Editing*** visual data to conform to a desired style, while ***preserving the underlying content***
- To propose expressing the desired style through ***natural language*** (a text prompt), similar to how a commissioned ***artist*** is provided a verbal or textual description of the desired work.

Introduction

Intuitive control the 3D object by using natural language

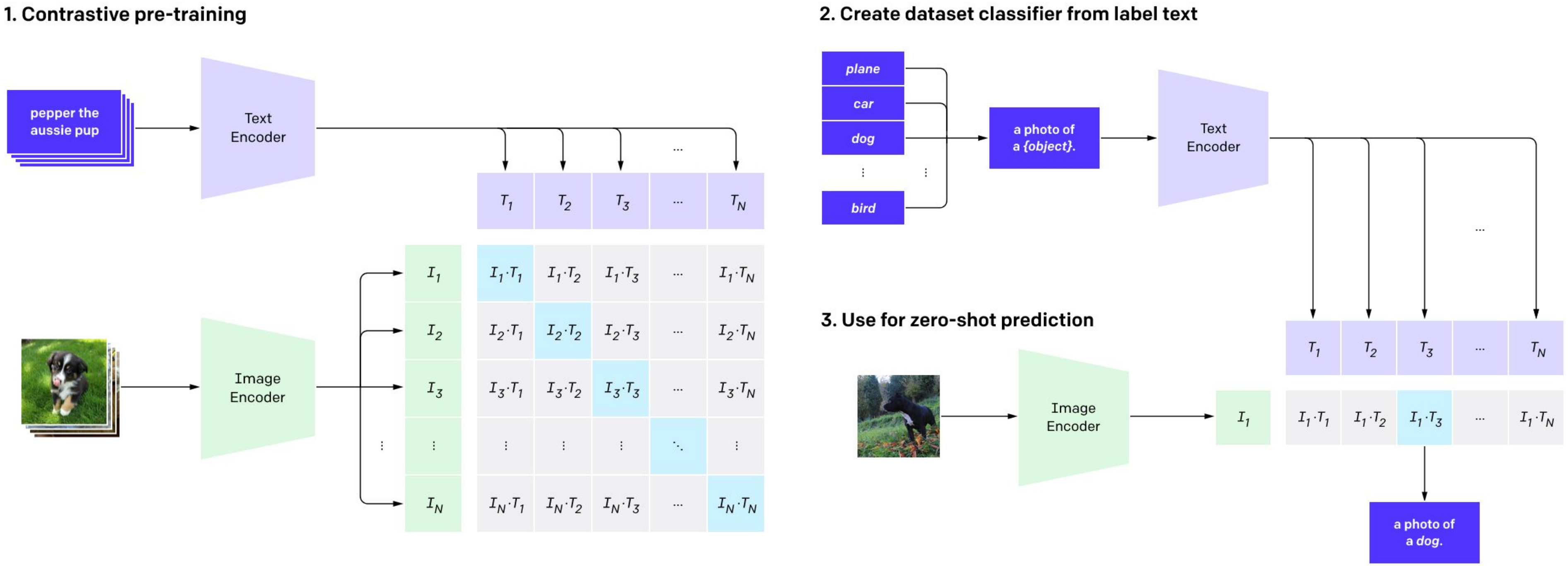


Style added

- ***Text2Mesh*** present a technique for the ***semantic manipulation of style*** for 3D meshes, harnessing the representational power of ***CLIP***.
- This system combines the ***advantages*** of ***explicit mesh surfaces*** and the ***generality of neural fields*** to facilitate intuitive control for stylizing 3D shapes.

CLIP (Contrastive Language–Image Pre-training)

CLIP builds on a large body of work on
zero-shot transfer, natural language supervision, and multimodal learning



CLIP (Contrastive Language–Image Pre-training) learns a joint embedding space for images and text

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021, July). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning* (pp. 8748-8763). PMLR.

Necessity of Network

Ablation on the priors used in Text2Mesh method (full) for a candle mesh and target 'Candle made of bark'

VS

full

-net
(without network)

However, a ***straightforward*** optimization of the 3D stylized mesh which maximizes the CLIP similarity score converges to a ***degenerate*** (i.e. noisy) ***solution*** → Employing ***CLIP*** for stylization requires ***careful regularization***

Related Work

- **Text driven manipulation:**
  This work to image manipulation techniques controlled through textual descriptions embedded by CLIP [1](Alex et.al). CLIP learns a joint embedding space for images and text, and other works have incorporate CLIP as means for text-guided image generation.

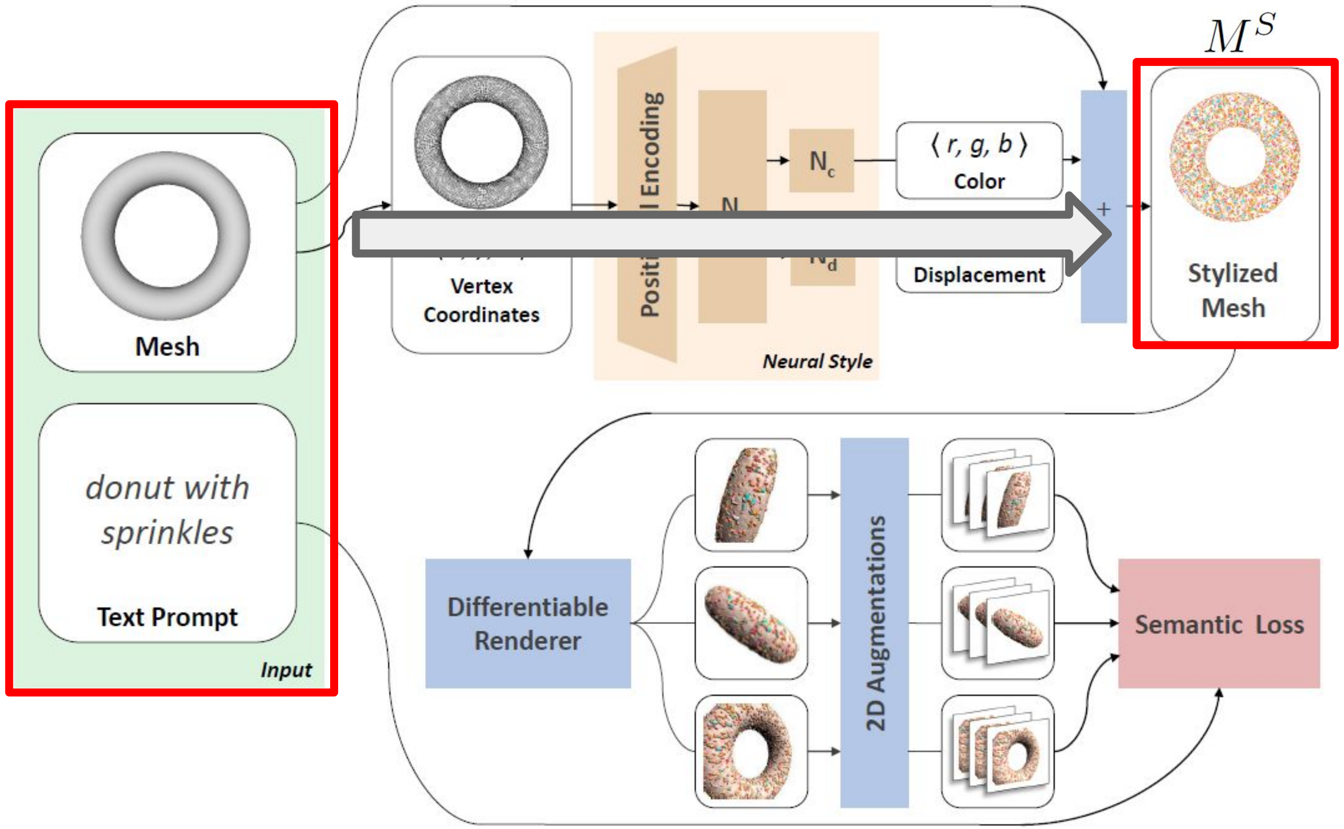
- **Geometric Style Transfer in 3D:**
  Some approaches analyze 3D shapes and identify similarly shaped geometric elements and parts which differ in style [2](Ruizhen et.al). Others transfer geometric style based on content/style separation [3](Xu et.al)

- **Texture Transfer in 3D:**
  Aspects of a 3D mesh style can be controlled by texturing a surface through mesh parameterization [4](Mark et.al), but recent work explored a neural representation of texture[5] (Nicholas et.al)

[1] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. arXiv preprint arXiv:2103.00020, 2021.
[2] Ruizhen Hu, Wenchao Li, Oliver Van Kaick, Hui Huang, Melinos Averkiou, Daniel Cohen-Or, and Hao Zhang. Colocating style-defining elements on 3d shapes. ACM Transactions on Graphics (TOG), 36(3):1–15, 2017.
[3]Xu Cao, Weimin Wang, Katashi Nagao, and Ryosuke Nakamura. Psnet: A style transfer network for point cloud stylization on geometry and color. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 3337–3345, 2020.
[4] Mark Gillespie, Boris Springborn, and Keenan Crane. Discrete conformal equivalence of polyhedral surfaces. ACM Transactions on Graphics (TOG), 40(4):1–20, 2021.
[5] Nicholas Sharp. Intrinsic Triangulations in Geometry Processing. PhD thesis, Carnegie Mellon University, August 2021.

Overall Architecture

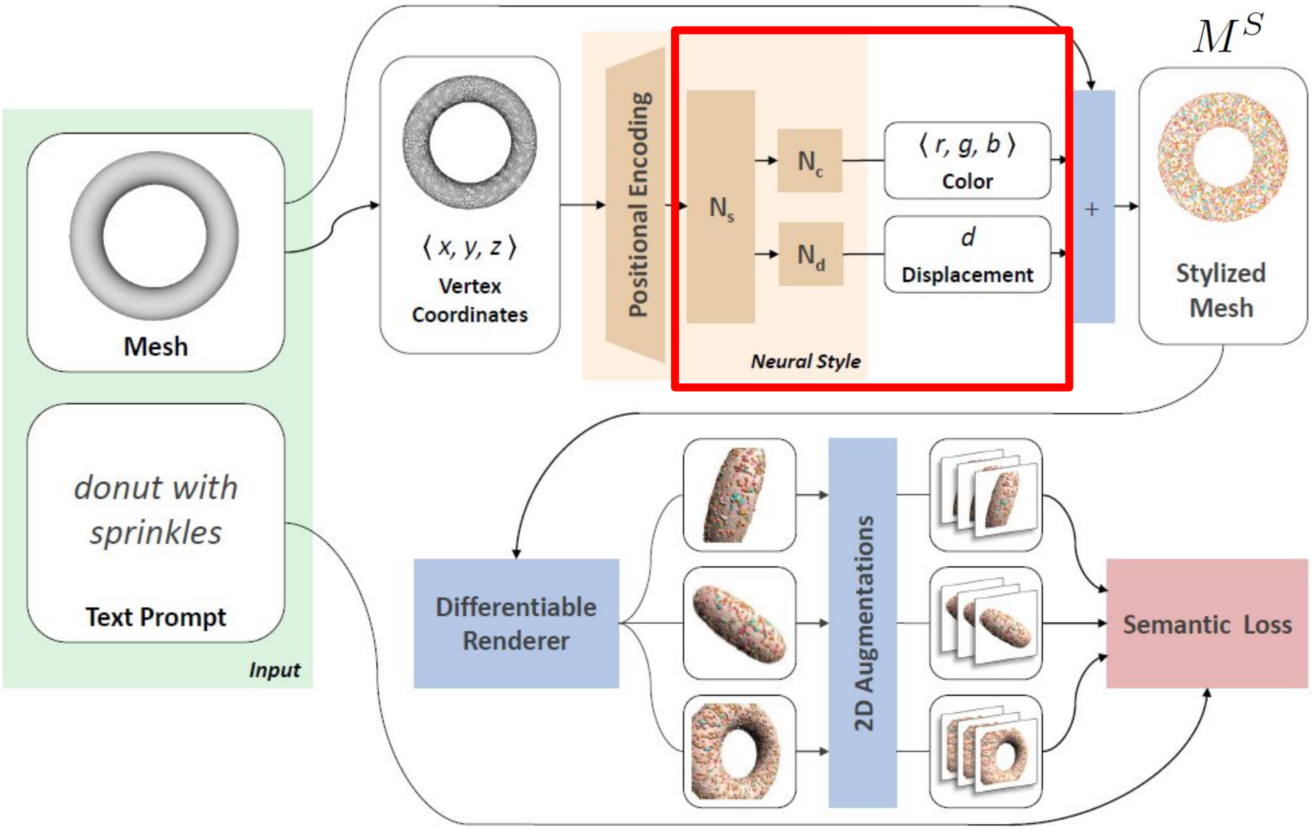
Figure 4.



- The object's style (color and local geometry) is ***modified*** to conform to a ***target text prompt*** t, resulting in a ***stylized mesh*** M^S

Overall Architecture

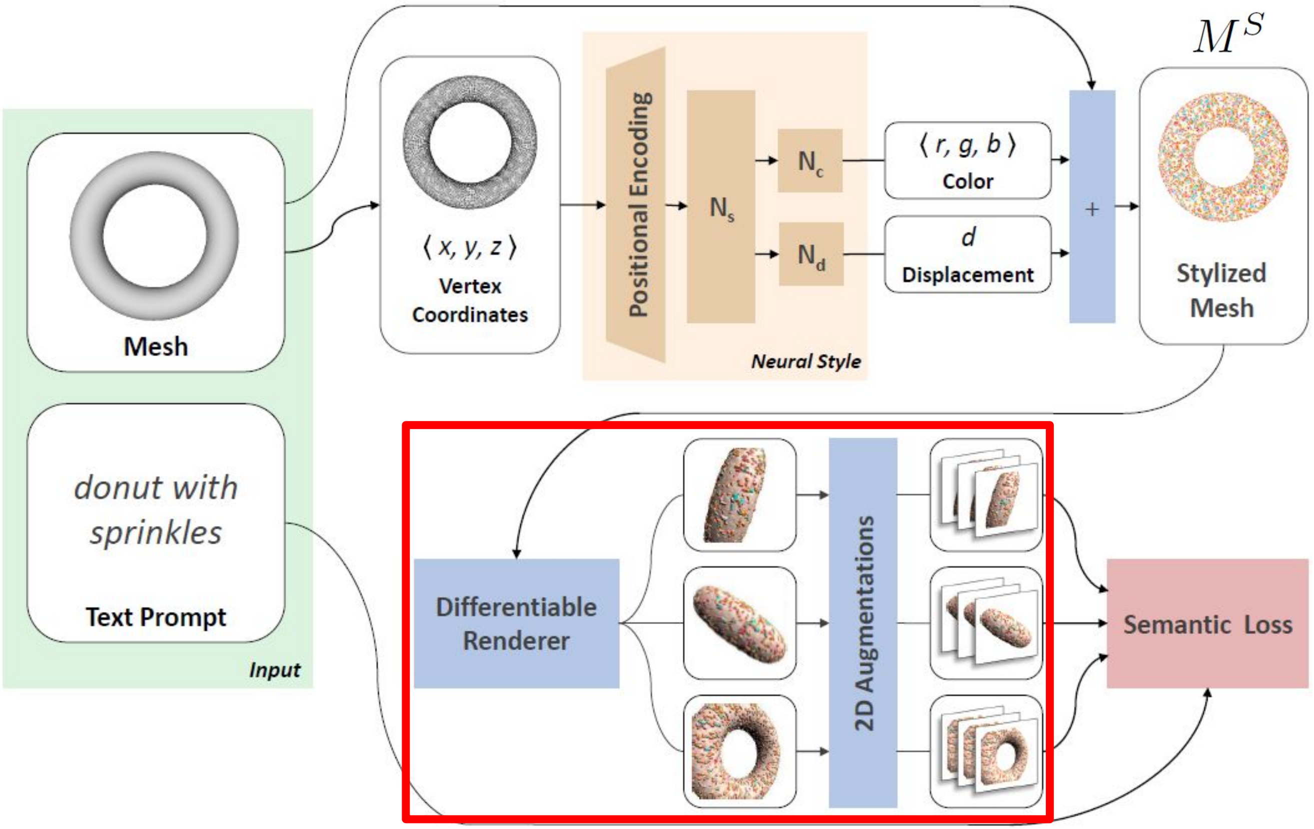
Figure 4.



- The ***NSF*** (Neural Style Field) Network ***learns to map points*** on the mesh surface to an ***RGB color and displacement*** along the normal direction.

Overall Architecture

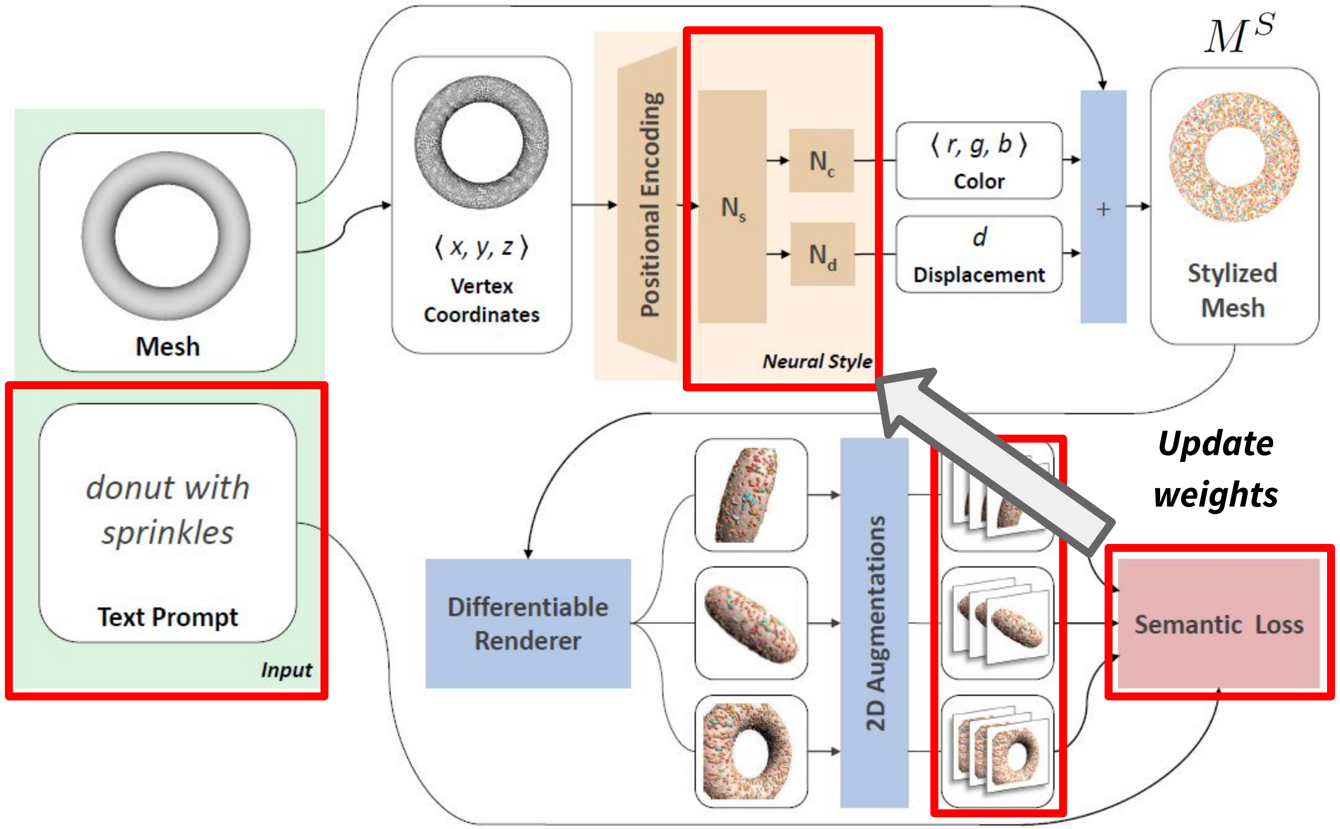
Figure 4.



- Render M^S from multiple views and apply 2D augmentations that are embedded using ***CLIP***.

Overall Architecture

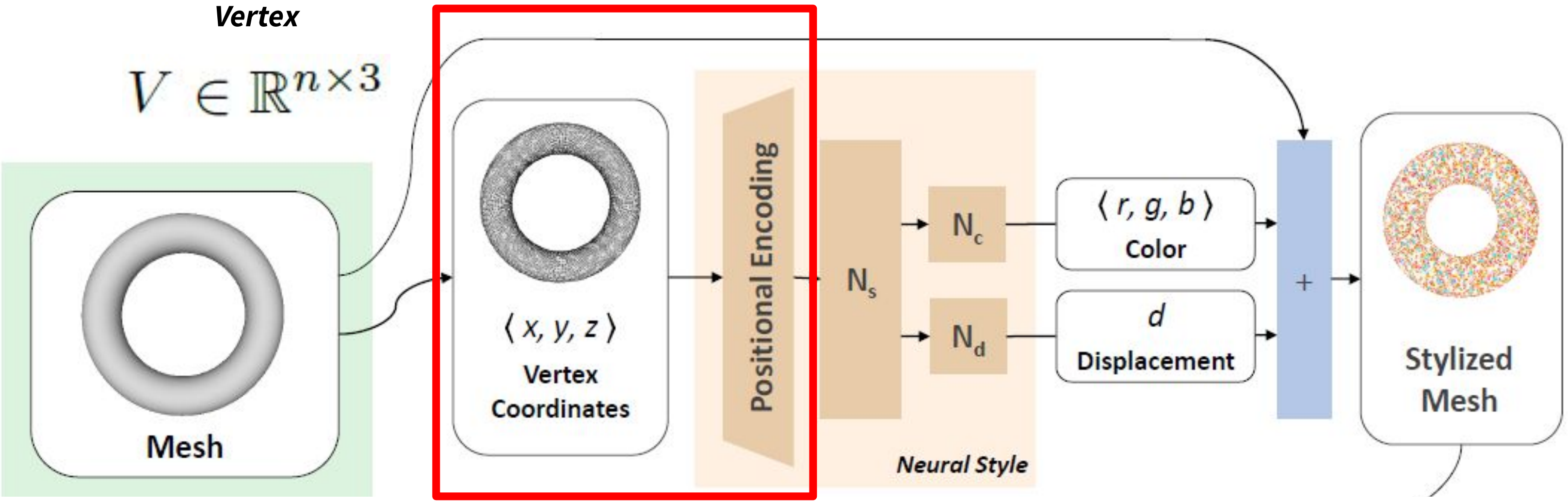
Figure 4.



- The ***CLIP similarity*** between the rendered and augmented ***images*** and the ***target text*** is used as a signal to ***update*** the neural network ***weights***.

Architecture

Figure 4.



Positional encoding features γ(p)

$$\gamma(p) = \left[\cos(2\pi \mathbf{B}p), \sin(2\pi \mathbf{B}p)\right]^{\mathrm{T}}$$

- Normalize the coordinates and map a vertex to a 256-dimensional Fourier feature
- Per-vertex ***positional encoding*** features *γ(p)* are passed as input to an MLP *Ns*, which then branches out to MLPs *Nd* and *Nc*

Positional Encoding

Figure 5.
Ablation on the priors used in our method (full) for a candle mesh and target 'Candle made of bark'

full: our method
−*net*: without our style field network
−*aug*: without 2D augmentations
−*FFN*: without positional encoding
−*crop*: without crop augmentations for *ψlocal*
−*displ*: without the geometry-only component of *Lsim*
−*3D*: learning over a 2D plane in 3D space

CLIP score: $\mathrm{sim}(\hat{S}^{\mathrm{full}}, \phi_{\mathrm{target}})$

Semantic loss: $\mathcal{L}_{\mathrm{sim}} = \sum_{\hat{S}} \mathrm{sim}\left(\hat{S}, \phi_{\mathrm{target}}\right)$

Utilize the *positional encodings* using fast *fourier feature networks* what enables us to obtain the fine grained results to solve the *spectral bias* problem

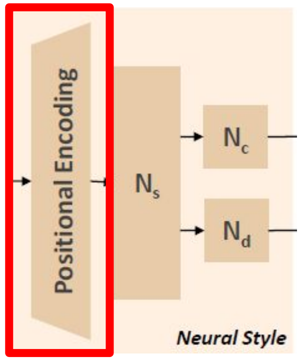
Positional Encoding

full

vs

-FFN

Figure 4.



Utilize the ***positional encodings*** using fast ***fourier feature networks*** what enables us to obtain the fine grained results to solve the ***spectral bias*** problem

Positional Encoding

σ: The amount of frequencies that are going into the positional encoding

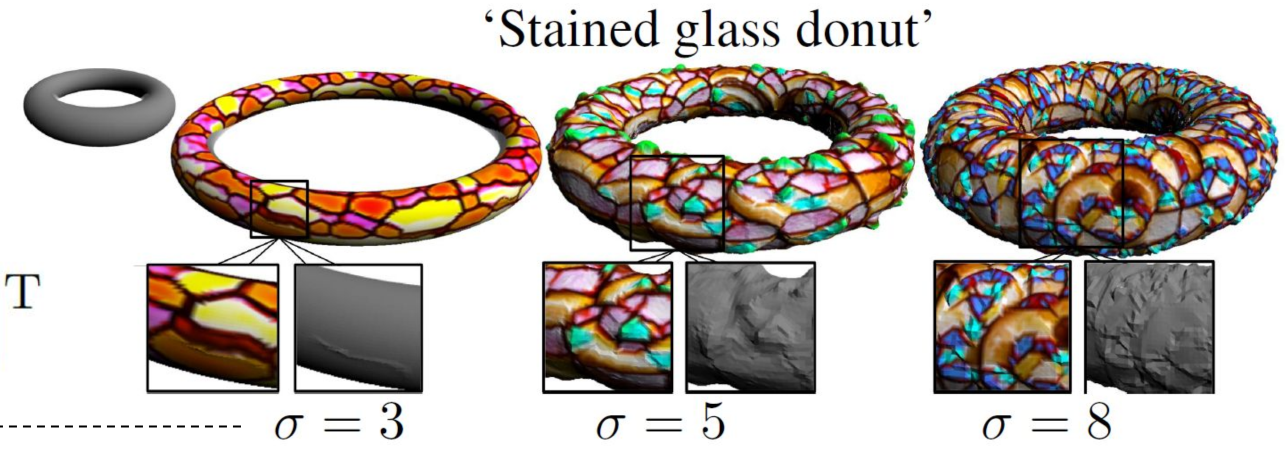


Figure 7. Increasing the range of input frequencies in the positional encoding using increasing SD σ for matrix $\mathbf{B}$ in Eq. (1).

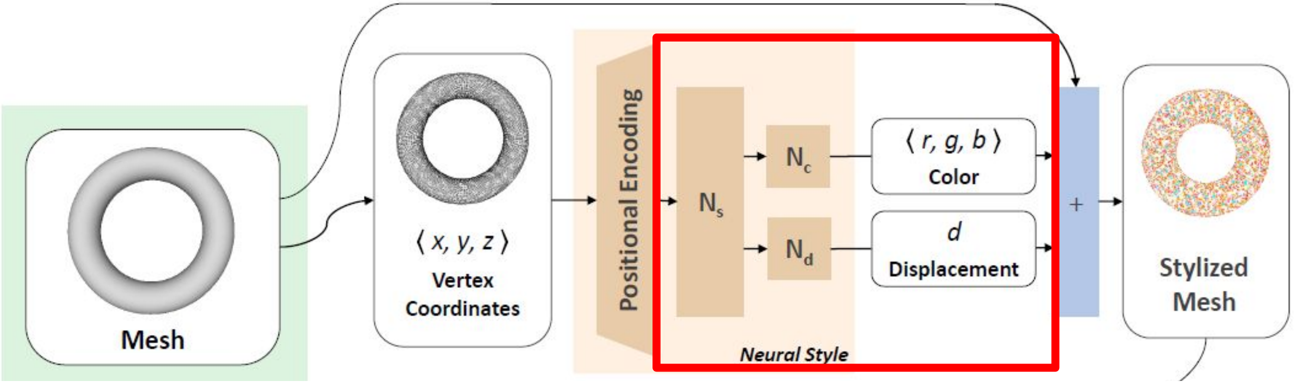
For every point p its positional encoding γ(p) is given by:

$$\gamma(p) = \left[\cos\left(2\pi \mathbf{B}p\right), \sin\left(2\pi \mathbf{B}p\right)\right]^{\mathrm{T}}$$

- The network leverages a ***positional encoding*** where the range of ***frequencies*** can be ***directly controlled*** by the standard deviation σ of the B matrix
- Increasing the frequency value increases the frequency of style details on the mesh and produces ***sharper*** and ***more frequent displacements*** along the normal direction

Architecture

Figure 4.



Four 256-dimensional linear layers
& ReLU activation

N_s

N_c N_d

Each of them have two 256-dimensional linear layers
& ReLU activation

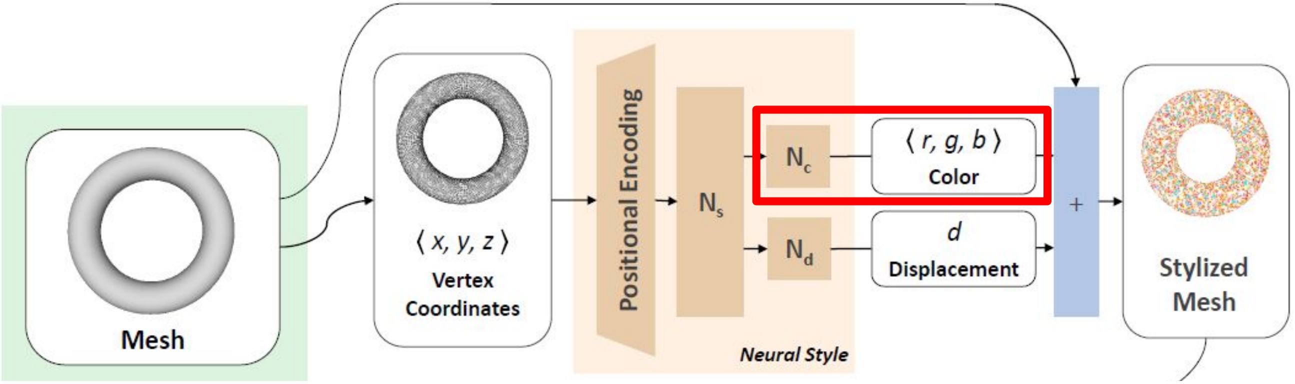
tanh activation

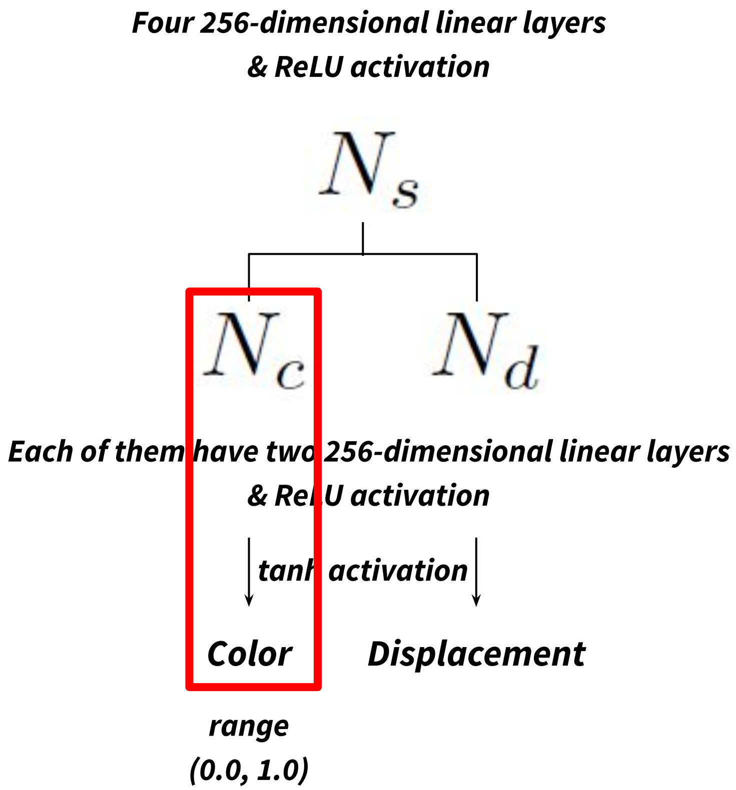
Color *Displacement*

- The shared MLP layers *Ns* consist of four 256-dimensional linear layers with ReLU activation.
- The branched layers, *Nd* and *Nc*, each consist of two 256-dimensional linear layers with ReLU activation.
- After the final linear layer, a tanh activation is applied to each branch.

Architecture

Figure 4.



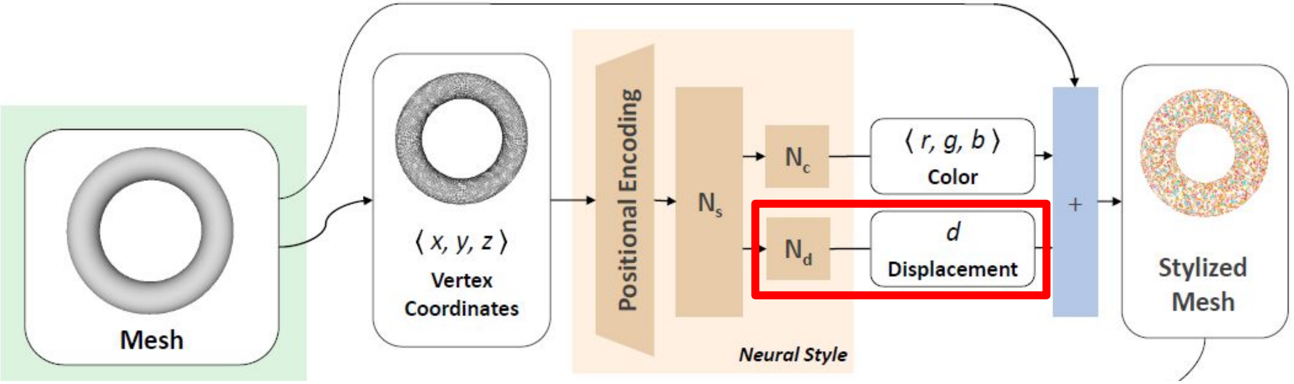


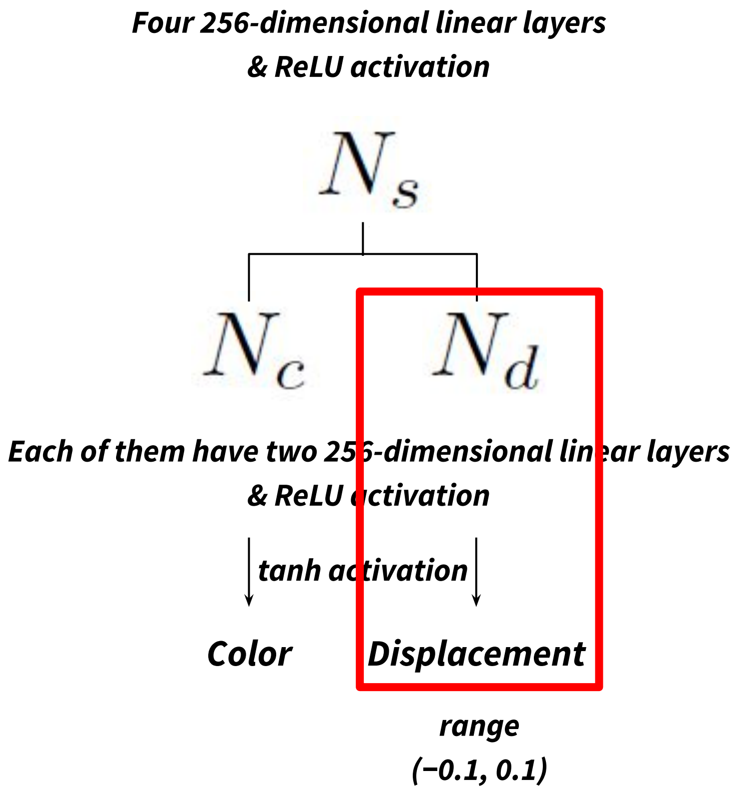
$$c_p \in [0,1]^3$$

- Divide the output of ***Nc*** by 2 and add it to [0.5, 0.5, 0.5]
- This enforces the final ***color prediction cp*** to be in range ***(0.0, 1.0)***

Architecture

Figure 4.

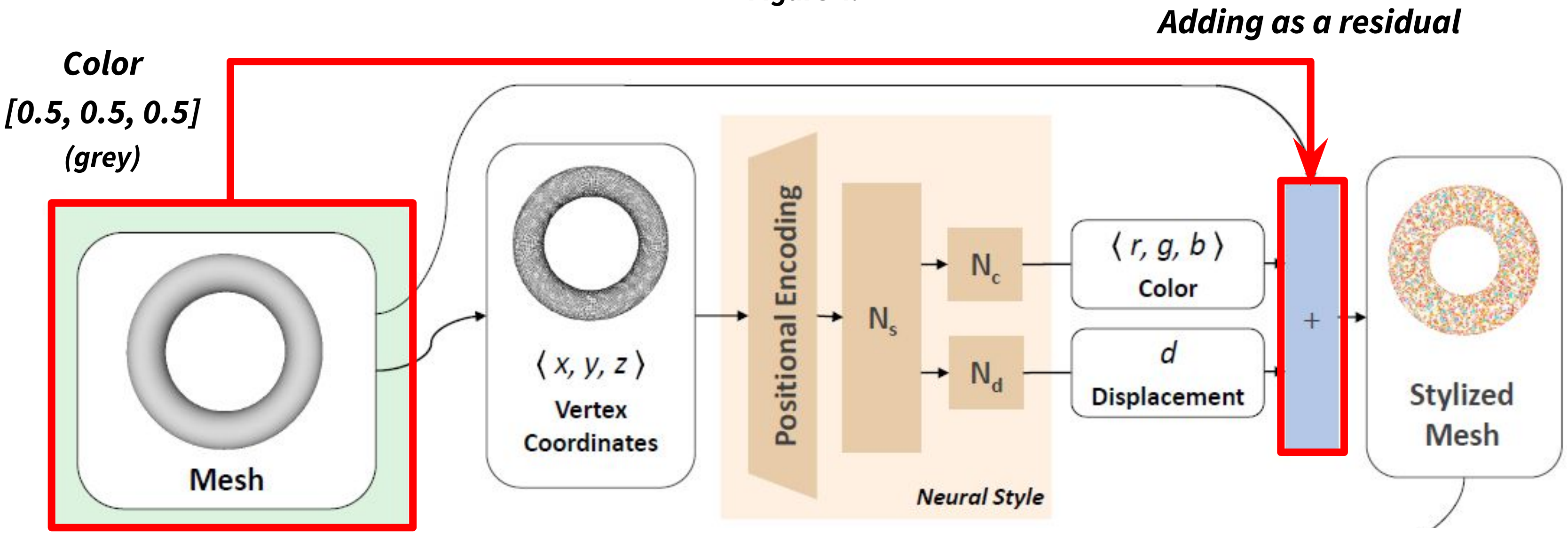




- For the branch ***Nd***, multiply the final tanh layer by 0.1 to get displacements in the range ***(−0.1, 0.1)***
- Constrain d_p to be in the range (−0.1, 0.1) to ***prevent content-altering displacements***

Architecture

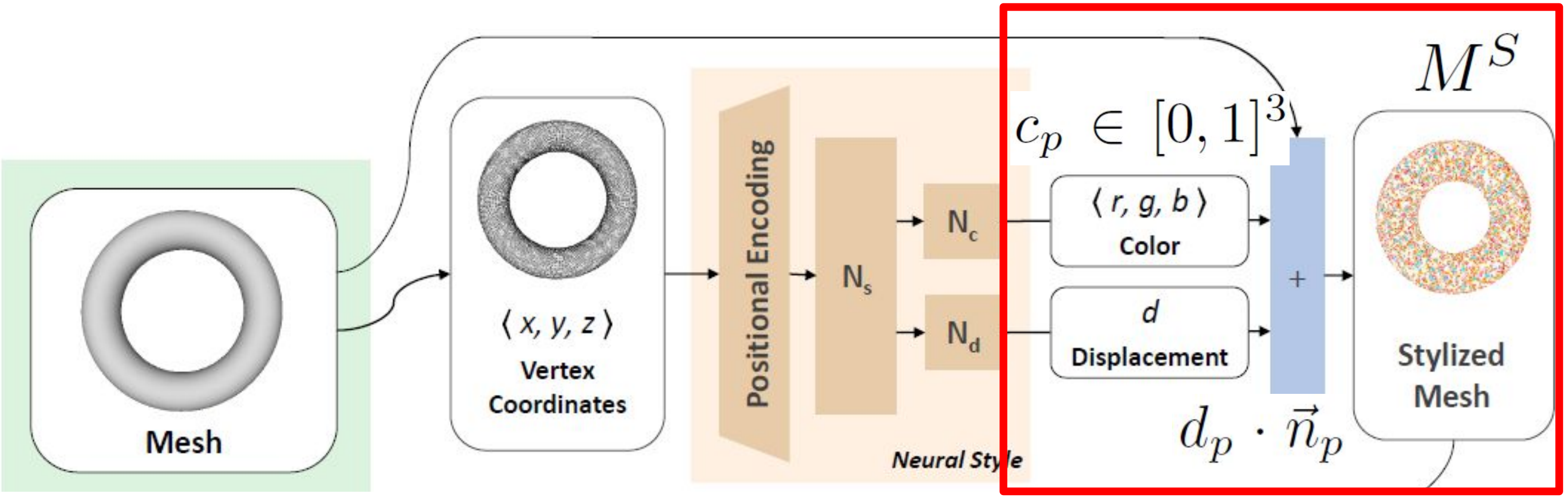
Figure 4.



Author find that initializing the ***mesh color to [0.5, 0.5, 0.5] (grey)*** and ***adding the network output as a residual*** helps ***prevent undesirable solutions*** in the early iterations of training.

Architecture

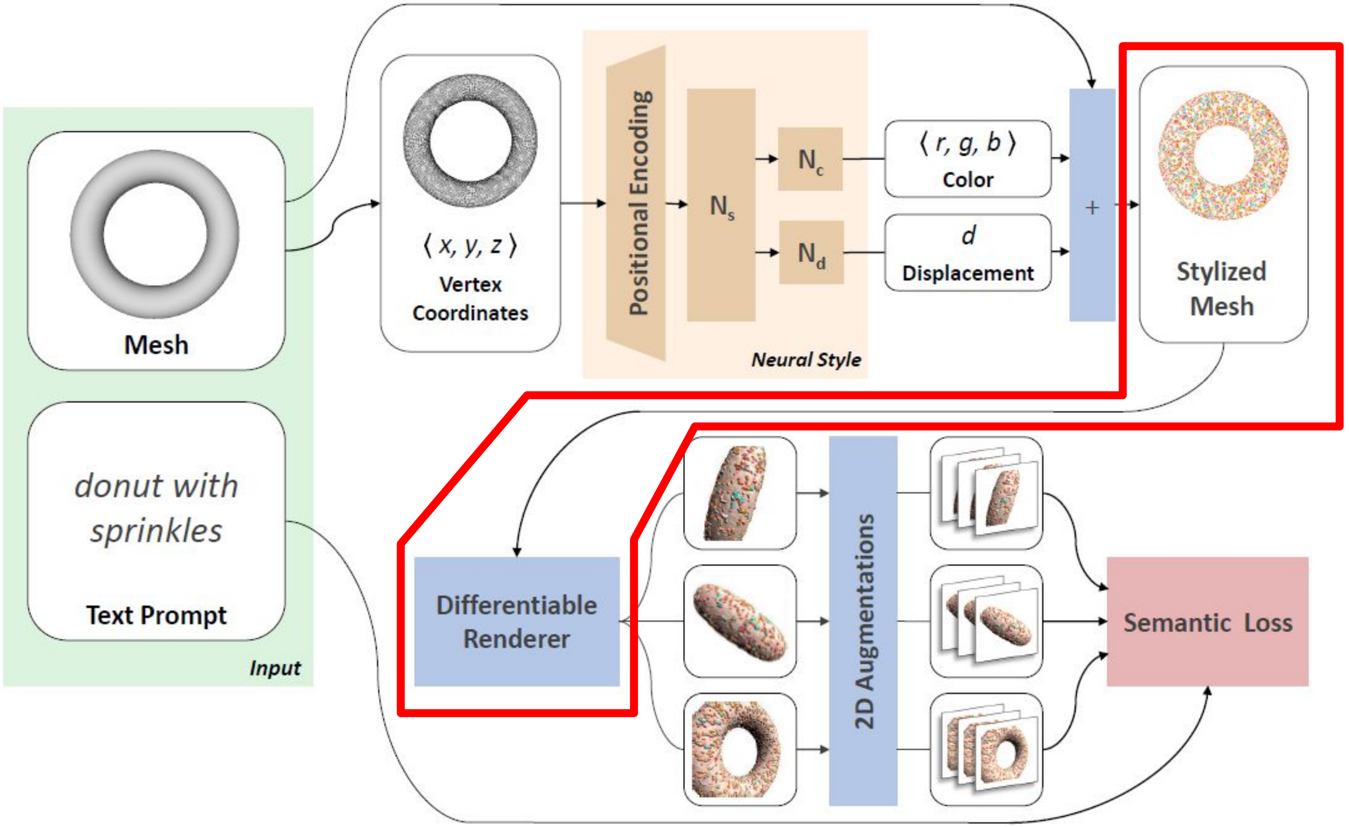
Figure 4.



- Every point *p* is displaced by $d_p \cdot \vec{n}_p$ and colored by *cp* to obtain stylized mesh prediction M^S
- ***Vertex colors propagate over the entire mesh surface*** using an interpolation-based differentiable renderer

Architecture

Figure 4.



Sample nθ *(= 5)* ***views*** around a predefined ***anchor view*** and render them using a ***differentiable renderer***, for given the stylized mesh M^S and the displaced mesh M^S_{displ}

Anchor View Choice

Views with Highest and lowest CLIP Score

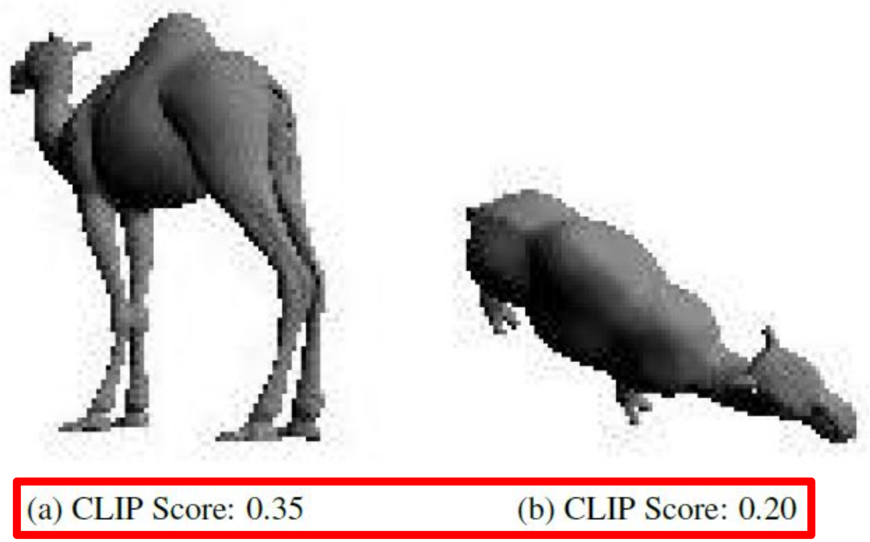


Figure 20. Example views with CLIP similarities.

CLIP score of the view that passes from the vertex to the center of the mesh

Figure 19. CLIP scores for each vertex view.

- Select the view with the ***highest*** (i.e. best) ***CLIP similarity*** to the content ***as the anchor*** which will allow a high-quality stylization
- This ***metric is limited in expressiveness***, however, as demonstrated by the ***constrained range*** that the scores fall within for all the views around the mesh.

Anchor View Choice

(a) # Views: 5 (Base)

(b) # Views: 6

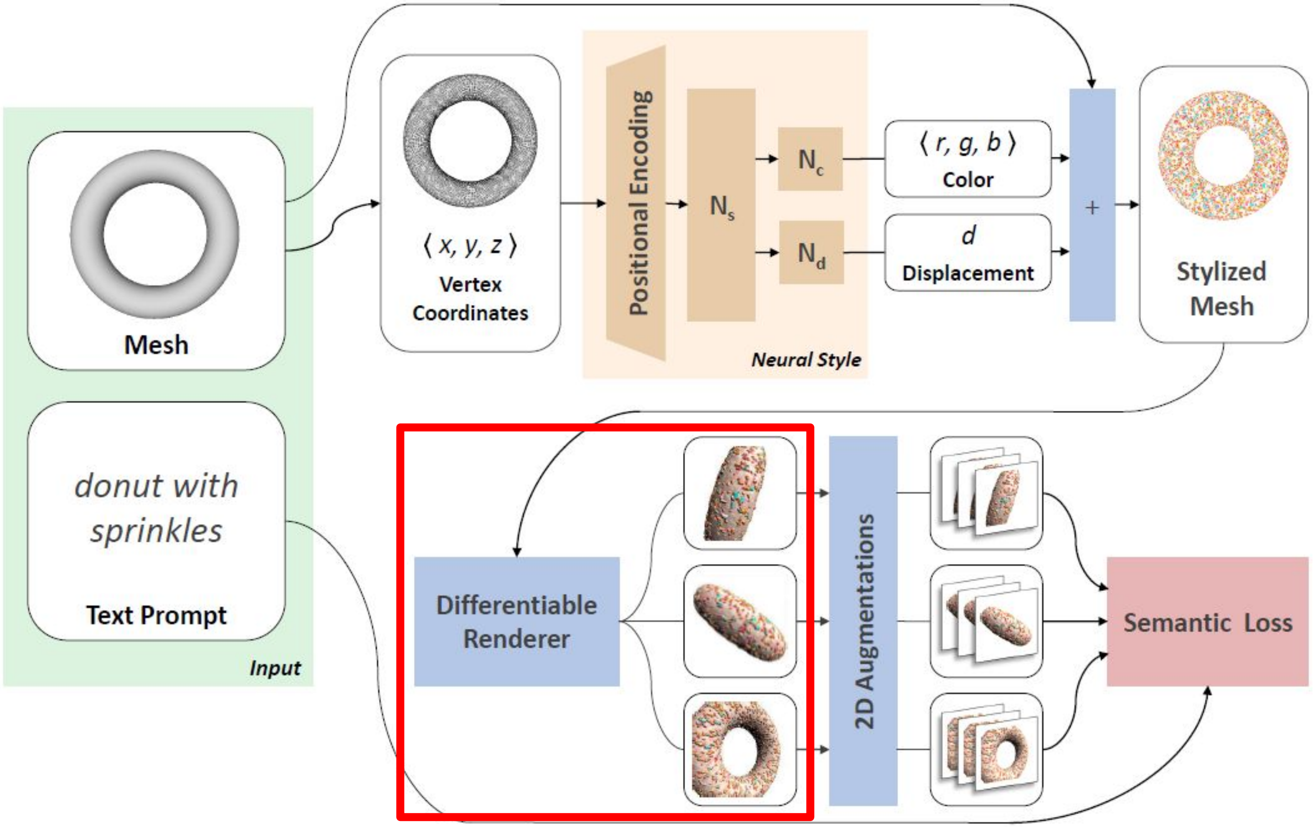
(c) # Views: 7

Figure 21. Style outputs sampling different # views. Prompt: 'A horse made of cactus'

Increasing the number of views ***beyond 5*** does ***little to change the quality*** of the output stylization

Architecture

Figure 4.



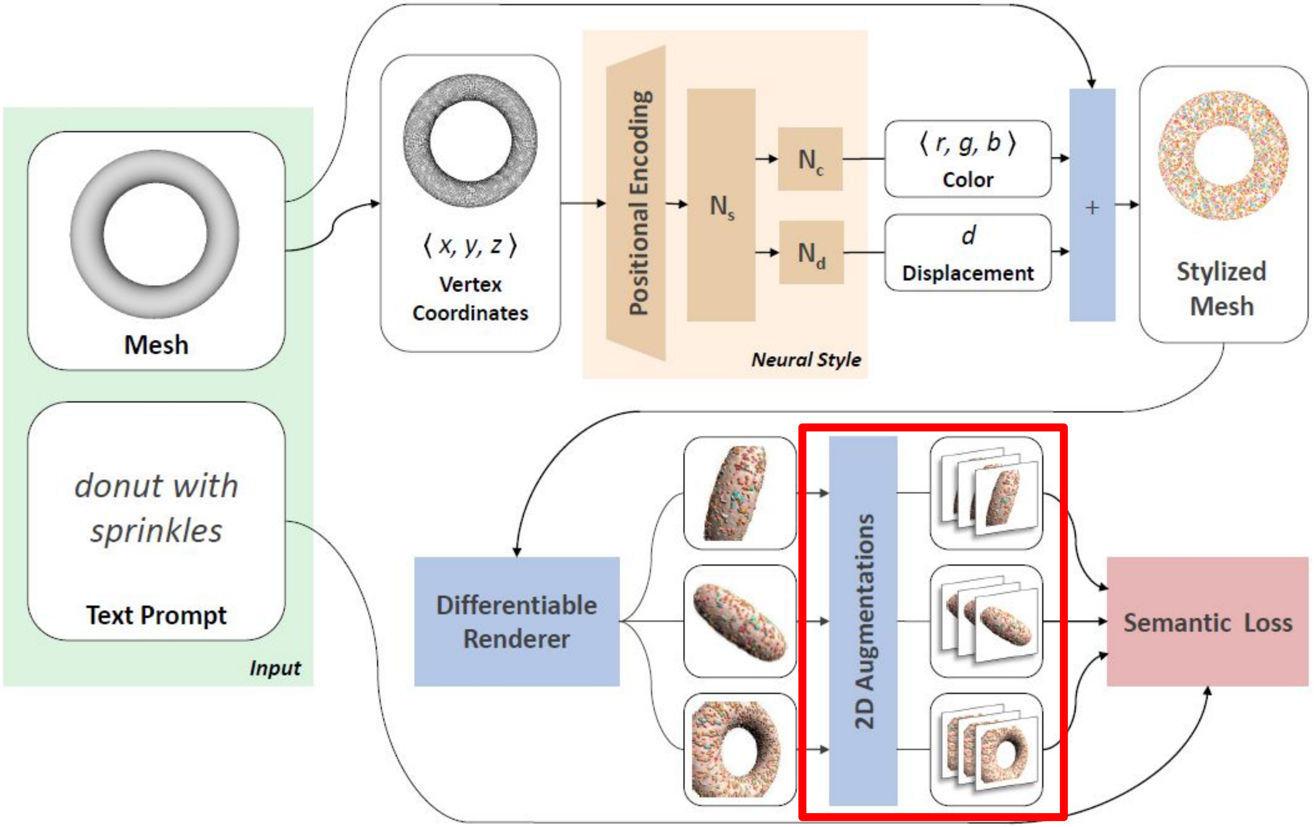
For each view, θ, render two 2D projections of the surface I_θ^{full} for M^S and I_θ^{displ} for M^S_{displ}

M^S : ***Stylized mesh***

M^S_{displ} : ***Displaced mesh***

Architecture

Figure 4.



- Draw a 2D augmentation *ψglobal* ∈ **Ψglobal** and *ψlocal* ∈ **Ψlocal**
- Apply *ψglobal, ψlocal* to the full view and *ψlocal* to the uncolored view, and embed them into CLIP space

2D augmentations (*ψglobal*, *ψlocal* and cropping)

ψglobal: Involves a ***random perspective*** transformation

Apply

Full View

Uncolored View

$$\psi_{\text{global}} \in \Psi_{\text{global}}$$

$$\psi_{\text{local}} \in \Psi_{\text{local}}$$

Embed

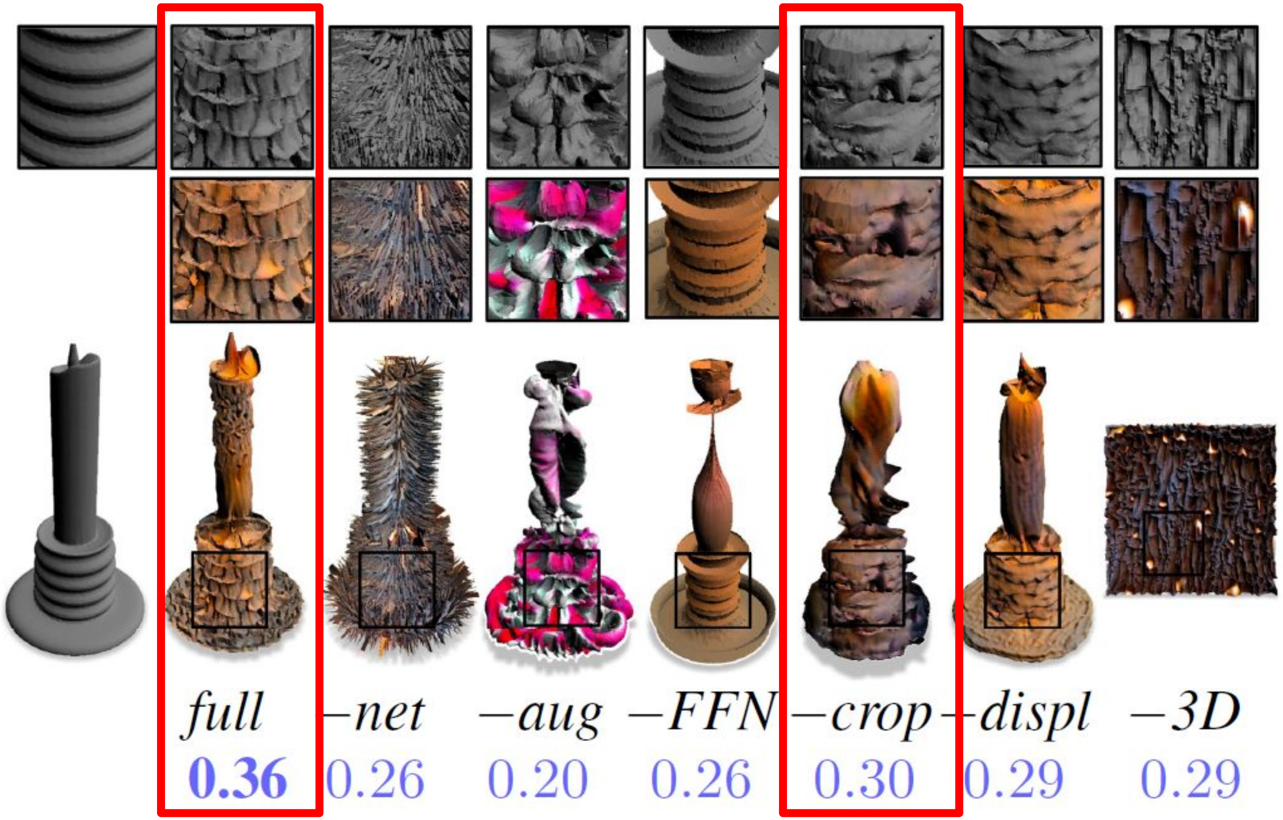
CLIP space

ψlocal: Generates both a ***random perspective*** and a ***random crop*** that is 10% of the original image

The 2D augmentations generated using ***ψglobal*** and ***ψlocal*** are critical for method to ***avoid degenerate solutions***

2D augmentations (*ψglobal*, *ψlocal* and cropping)

Figure 5.
Ablation on the priors used in our method (full) for a candle mesh and target 'Candle made of bark'



full: our method
−*net*: without our style field network
−*aug*: without 2D augmentations
−*FFN*: without positional encoding
−*crop*: without crop augmentations for *ψlocal*
−*displ*: without the geometry-only component of *Lsim*
−*3D*: learning over a 2D plane in 3D space

CLIP score: $\mathrm{sim}(\hat{S}^{\mathrm{full}}, \phi_{\mathrm{target}})$

Semantic loss: $\mathcal{L}_{\mathrm{sim}} = \sum_{\hat{S}} \mathrm{sim}\left(\hat{S}, \phi_{\mathrm{target}}\right)$

Cropping **allows the network to focus on localized regions when making *fine grained adjustments* to the surface geometry and color (Check the -*crop*)**

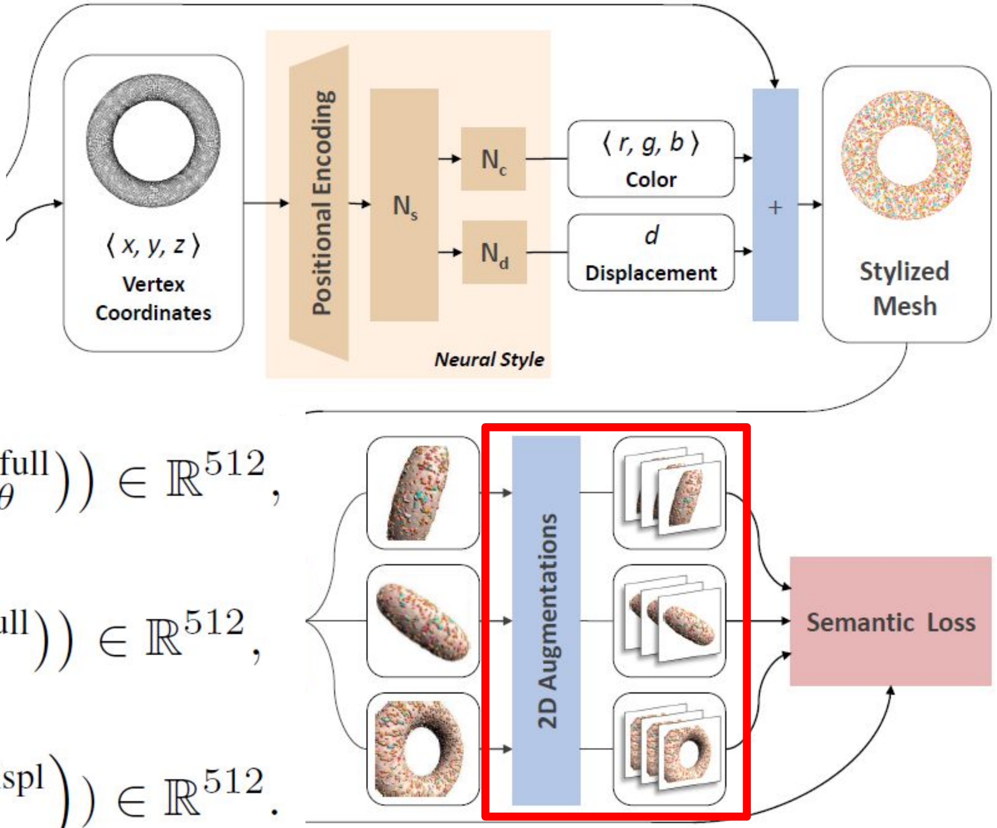
2D augmentations (*ψglobal*, *ψlocal* and cropping)

full *-crop*

Cropping allows the network to focus on localized regions when making ***fine grained adjustments*** to the surface geometry and color (Check the *-crop*)

Architecture

Figure 4.



M^S *: Stylized mesh*

M^S_{displ} *: Displaced mesh*

I_θ^{full} *: Surface for* M^S

I_θ^{displ} *: Surface for* M^S_{displ}

Average the embeddings

$$\hat{S}^{\text{full}} = \frac{1}{n_\theta} \sum_\theta E\left(\psi_{\text{global}}\left(I_\theta^{\text{full}}\right)\right) \in \mathbb{R}^{512},$$

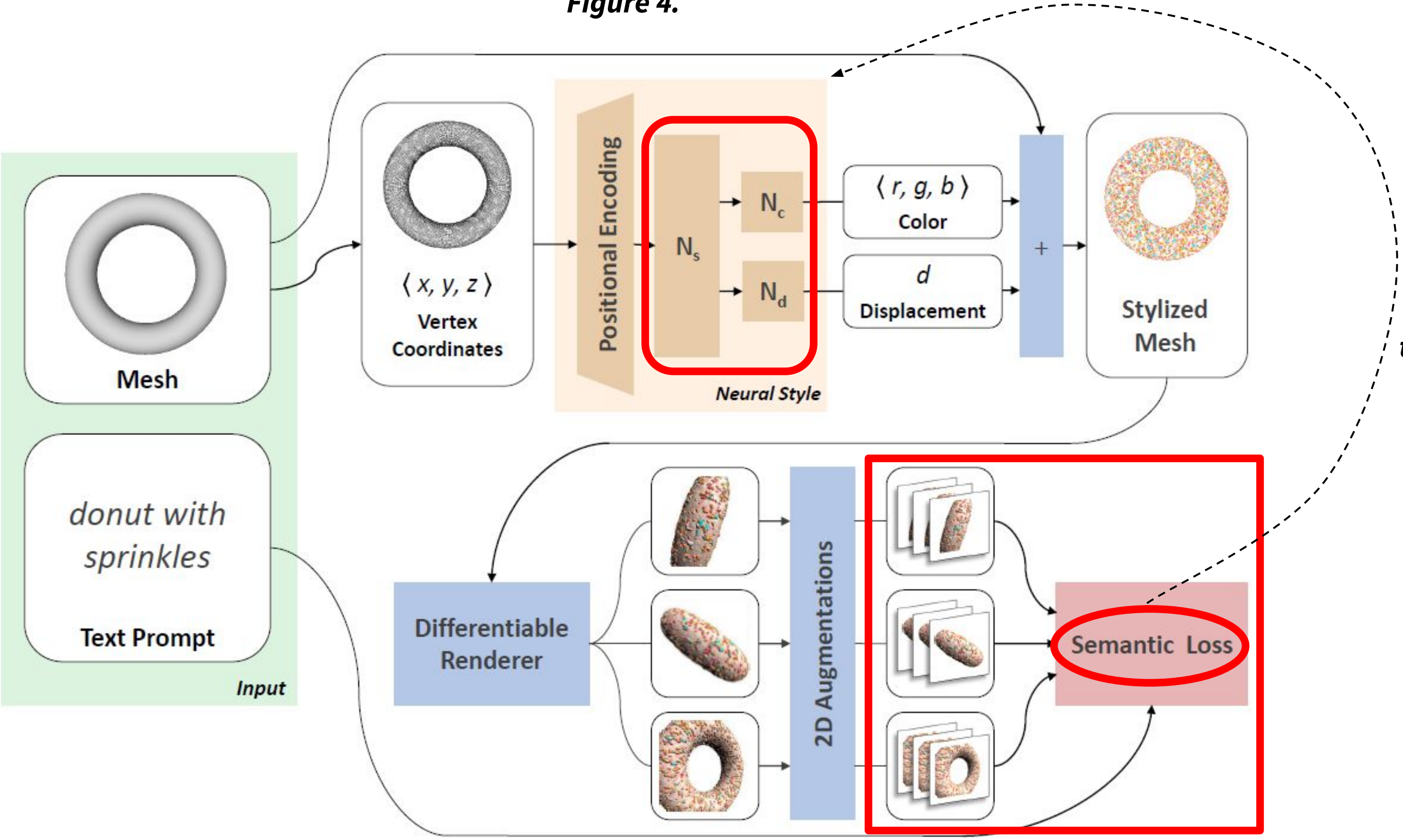
$$\hat{S}^{\text{local}} = \frac{1}{n_\theta} \sum_\theta E\left(\psi_{\text{local}}\left(I_\theta^{\text{full}}\right)\right) \in \mathbb{R}^{512},$$

$$\hat{S}^{\text{displ}} = \frac{1}{n_\theta} \sum_\theta E\left(\psi_{\text{local}}\left(I_\theta^{\text{displ}}\right)\right) \in \mathbb{R}^{512}.$$

- Average the embeddings across all views
- Consider an augmented representation of our input mesh as the average of its encoding from multiple augmented views

Architecture

Figure 4.



update
the weights of NSF Network

- **The target *t* is embedded through CLIP by** $\phi_{\text{target}} = E(t) \in \mathbb{R}^{512}$
- **Semantic Loss:**

$$\mathcal{L}_{\text{sim}} = \sum_{\hat{S}} \text{sim}\left(\hat{S}, \phi_{\text{target}}\right)$$

$$\hat{S} \in \{\hat{S}^{\text{full}}, \hat{S}^{\text{displ}}, \hat{S}^{\text{local}}\}$$

$$\text{sim}(a, b) = \frac{a \cdot b}{|a| \cdot |b|}$$

sim (a, b) is the cosine similarity between a and b

Updating the Weights of NSF Network

M^S *: Stylized mesh* I_θ^{full} *: Surface for* M^S

M^S_{displ} *: Displaced mesh* I_θ^{displ} *: Surface for* M^S_{displ}

Average the embeddings

$$\hat{S}^{\text{full}} = \frac{1}{n_\theta}\sum_\theta E\left(\psi_{\text{global}}\left(I_\theta^{\text{full}}\right)\right) \in \mathbb{R}^{512},$$

$$\hat{S}^{\text{local}} = \frac{1}{n_\theta}\sum_\theta E\left(\psi_{\text{local}}\left(I_\theta^{\text{full}}\right)\right) \in \mathbb{R}^{512},$$

$$\hat{S}^{\text{displ}} = \frac{1}{n_\theta}\sum_\theta E\left(\psi_{\text{local}}(I_\theta^{\text{displ}})\right) \in \mathbb{R}^{512}.$$

Semantic loss

$$\mathcal{L}_{\text{sim}} = \sum_{\hat{S}} \operatorname{sim}\left(\hat{S}, \phi_{\text{target}}\right)$$

$\hat{S}^{\text{full}}$

$\hat{S}^{\text{local}}$

Update

All

$N_s \quad N_c \quad N_d$

N_c → Color $c_p \in [0,1]^3$

N_d → Displacement along the vertex normal d_p

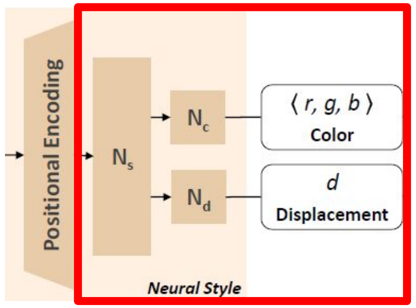
$\hat{S}^{\text{displ}}$

Update

$N_s \quad N_d$

Unrelated to Color

Figure 4.



$\hat{S}^{\text{full}}$ and $\hat{S}^{\text{local}}$ update *Ns*, *Nc* and *Nd* while **$\hat{S}^{\text{displ}}$ only updates *Ns* and *Nd***

Separation (geo-only loss | geo-and-color loss)

Figure 5.
Ablation on the priors used in our method (full) for a candle mesh and target 'Candle made of bark'

full: our method
−*net*: without our style field network
−*aug*: without 2D augmentations
−*FFN*: without positional encoding
−*crop*: without crop augmentations for *ψlocal*
−*displ*: without the geometry-only component of *Lsim*
−*3D*: learning over a 2D plane in 3D space

CLIP score: $\mathrm{sim}(\hat{S}^{\mathrm{full}}, \phi_{\mathrm{target}})$

Semantic loss: $\mathcal{L}_{\mathrm{sim}} = \sum_{\hat{S}} \mathrm{sim}\left(\hat{S}, \phi_{\mathrm{target}}\right)$

The ***separation*** into a ***geometry-only loss and geometry-and-color loss*** is an effective tool for ***encouraging meaningful changes in geometry*** (check the −*displ*)

Separation (geo-only loss | geo-and-color loss)

vs

full | *−displ*

The ***separation*** into a ***geometry-only loss and geometry-and-color loss*** is an effective tool for ***encouraging meaningful changes in geometry*** (check the −*displ*)

Experiments and Results

Input source meshes

Sources: COSEG, Thingi10K, Shapenet, Turbo Squid, and ModelNet

Features: Average of 79,366 faces, 16% non-manifold edges, 0.2% non-manifold vertices, and 12% boundaries

Experiments and Results

Figure 2.

Maintaining global semantics and preserving the underlying content

Experiments and Results

Figure 1.

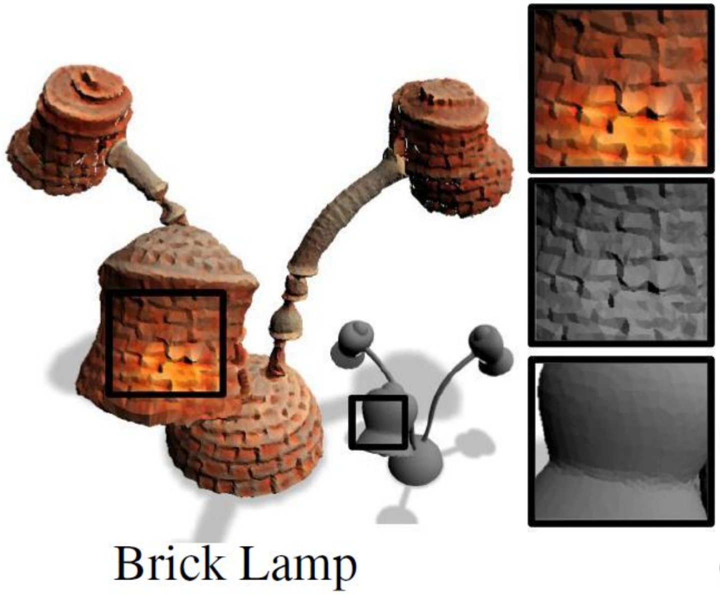


Figure 2.

Generates structured textures which are aligned to sharp curves and features

Experiments and Results

Figure 6. Our neural texture field stylizes the entire 3D shape.

Styles the entire mesh in a consistent manner that is part-aware and exhibits natural variation in texture

Experiments and Results

σ: The amount of frequencies that are going into the positional encoding

For every point p its positional encoding γ(p) is given by:

$$\gamma\left(p\right)=\left[\cos\left(2\pi\mathbf{B}p\right),\sin\left(2\pi\mathbf{B}p\right)\right]^{\mathrm{T}}$$

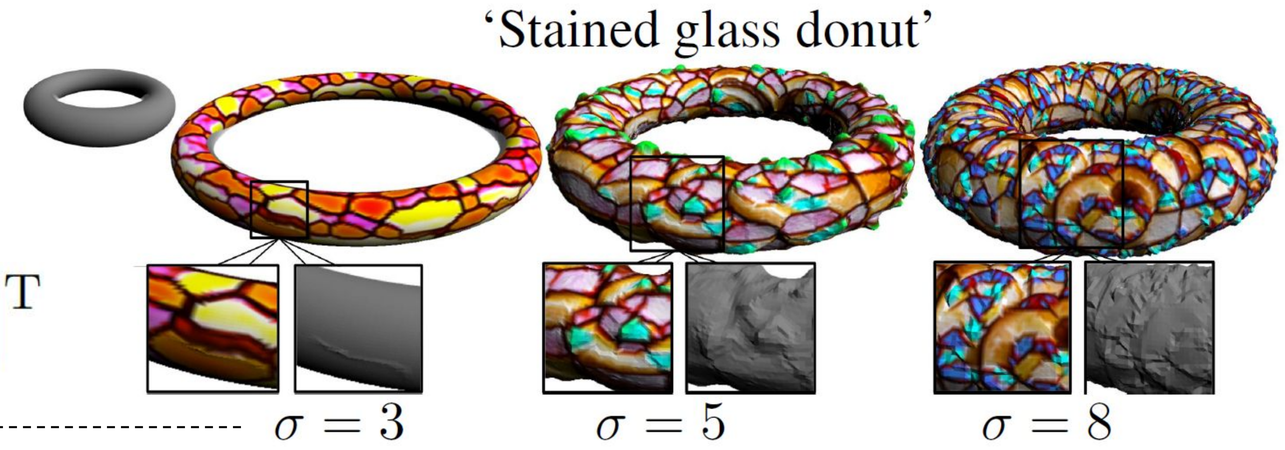
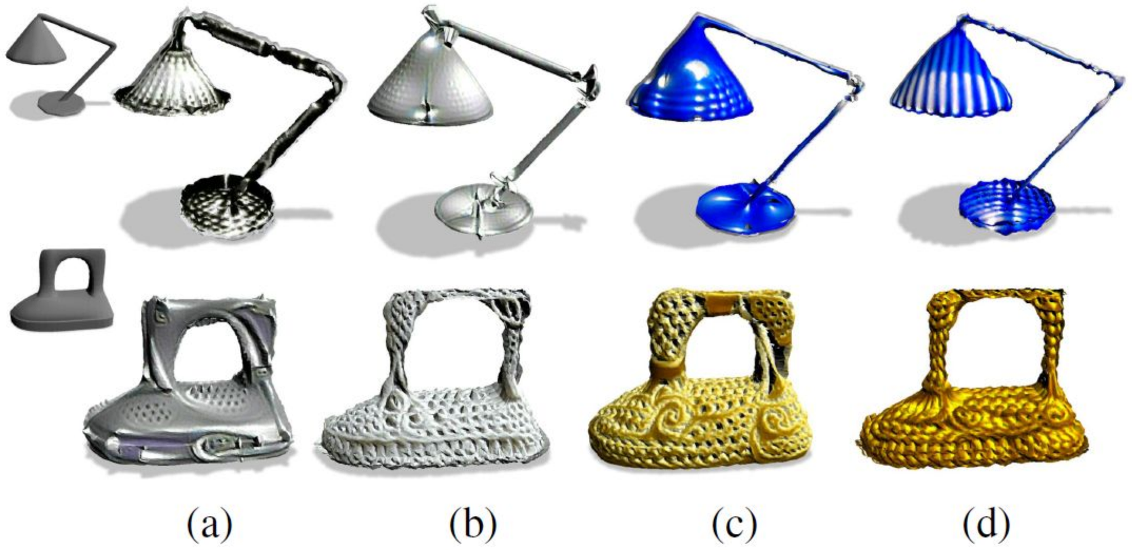


Figure 7. Increasing the range of input frequencies in the positional encoding using increasing SD σ for matrix $\mathbf{B}$ in Eq. (1).

Increasing the **σ** (frequency value) increases the frequency of style details on the mesh and produces ***sharper*** and ***more frequent displacements*** along the normal direction

Experiments and Results

Figure 8.
Increasing the target text prompt granularity for a source mesh of a lamp and iron.



Increasing style details

(a). ‘Lamp’, (b). ‘Luxo lamp’, (c). ‘Blue steel luxo lamp’, (d). ‘Blue steel luxo lamp with corrugated metal.
(a). ‘Clothes iron’, (b). ‘Clothes iron made of crochet’, (c). ‘Golden clothes iron made of crochet’, (d). ‘Shiny golden clothes iron made of crochet’.

- **Successfully synthesize styles of varying levels of specificity.**
- **Retention of the style details from each level of target granularity to the next.**

Experiments and Results

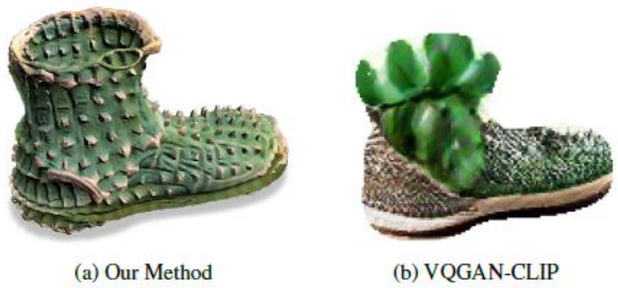


Figure 22. Prompt: 'A shoe made of cactus'

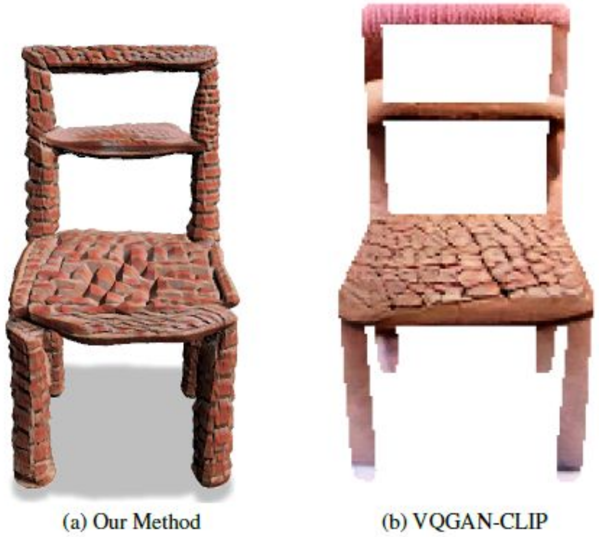


Figure 23. Prompt: 'A chair made of brick'

VQGAN-CLIP:
Synthesizes color inside a binary 2D mask projected from the 3D source shape (without 3D deformations) guided by CLIP

	(Q1): Overall	(Q2): Content	(Q3): Style
VQGAN	2.83 ($\pm$0.39)	3.60 ($\pm$0.68)	2.59 ($\pm$0.44)
Ours	**3.90** ($\pm$0.37)	**4.04** ($\pm$0.53)	**3.91** ($\pm$0.51)

Table 1. Mean opinion scores (1-5) for Q1-Q3 (see Sec. 4.3), for our method and baseline (control score: 1.16).

57 users evaluate 8 samples

(Q1) "How natural is the output depiction of {content} + {style}?"

(Q2) "How well does the output match the original {content}?"

(Q3) "How well does the output match the target {style}?"

Outperforms the VQGAN baseline across all questions, with a difference of 1.07, 0.44, and 1.32 for Q1-Q3

Limitation

Figure 15.
Geometric content and target style synergy.

- ***Assumes*** there ***exists*** a ***synergy*** between the ***input 3D geometry*** and the ***target style prompt***.
- However, stylizing a 3D mesh (e.g., dragon) towards an ***unrelated/unnatural prompt*** (e.g., stained glass) may result in a stylization that ***ignores the geometric prior*** and ***erases the source shape content***.

Limitation

Figure 15.
Geometric content and target style synergy.

The author ***solve*** this by simply ***including the object category in the text prompt*** (e.g., stained glass dragon) which adds a content preservation constraint into the target.

Limitation

Figure 25. Our method enables visualizing the biases in the CLIP embedding space. Given a human male input (source in Figure 3), and target prompt: 'a nurse', we observe a gender bias in CLIP to favor female shapes.

- ***Societal bias***: The nurse style in Fig. 25 is biased towards adding female features to the input male shape.
- Present in joint image-text embeddings through our stylization framework

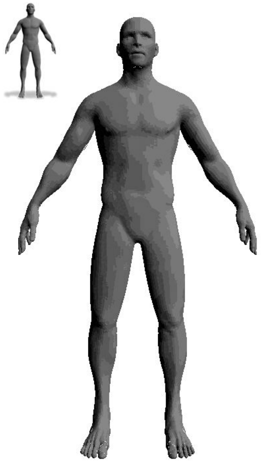
Conclusion

Mesh stylization through 2D projections

Key Points

1. Intuitive control over 3D shape manipulation
2. Without a directional field or mesh parameterization
3. Without relying on a pre-trained GAN network or a 3D dataset

a vase made of colorful crochet



Thank you.