Enhancing 3D Fidelity of Text-to-3D using Cross-View Correspondences

Supplementary Material

In this supplementary material, we provide additional details and qualitative results of CorrespondentDream which were not included in the main paper due to spatial constraints. We detail the refining and post-processing of correspondences in Appendix A, and explain additional implementation details of CorrespondentDream in Appendix B. In Appendix C, we elaborate on the effect the value of CFG (ω) has on the generated 3D object, and devise a CFG scheduling scheme to yield more satisfying results. We visualize the cross-view correspondences in Appendix D. We analyze the results when using cross-view correspondence loss as a pre-processing or post-processing method in Appendix E. We demonstrate that even at reduced image resolutions used in CorrespondentDream, the quality and 3D infidelities of the outputs are maintained in Appendix F. We visualize the intermediate rendered outputs of CorrespondentDream in comparison to the baseline (MVDream [28]) in Appendix G. We elaborate on the applicability of CorrespondentDream on other zero-shot text-to-3D generation methods in Appendix H. We outline the computation cost of CorrespondentDream, and evidence that the improved 3D fidelity does not come from the additional computation in Appendix I. We demonstrate that using off-the-shelf image matchers give dissatisfactory results, and substantiate the use of annotation-free cross-view correspondences in Appendix J. Finally, we present some example text prompts used for our experiments in Appendix K.

A. Correspondence post-processing details

In Sec. 4.3, we mentioned that the correspondences are determined from the correlation map as shown in Eq. (7). In this section, we describe the refining and post-processing steps which were applied to the correlation map and the sampled correspondences to yield a more robust set of correspondences that are aligned with human common sense.

Motivation. As we do not have the ground-truth correspondences between rendered views, we couldn't directly evaluate if our cross-view correspondences align sufficiently well with human perception in the presence of 3D infidelities. To this end, we devised an indirect protocol to ensure that our annotation-free cross-view correspondences are consistent with human perception.

We sampled 10 text-to-3D outputs which seems to have near-perfect 3D fidelity by human eyes. This would mean that the NeRF depths are also consistent with human perception, allowing us to consider the NeRF reprojections as the pseudo-ground truth correspondences. We experimented with various refinement / post-processing settings to maximize the precision and recall of our annotation-free cross-view correspondences with respect to the pseudo-ground truth correspondences.

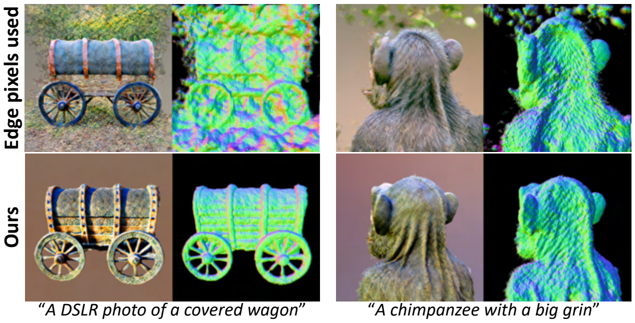
Edge pixels used
Ours
"A DSLR photo of a covered wagon"
"A chimpanzee with a big grin"

Figure A1. **Ablation of CorrespondentDream opacity-based foreground edge discarding**. It can be seen that not discarding the foreground edge pixels during the cross-view correspondence loss results in unwanted artifacts around the final output.

1. Filtering out by opacity. To ensure that we are establishing correspondences only between the foreground objects, *i.e.*, not the background, we use the NeRF opacity values to filter out the background positions. The opacity value ranges from 0 to 1, where 0 signifies no occupancy (background) and values closer to 1 signifies high occupancy (likely to be foreground). We can therefore disregard the background positions by filtering out pixel positions with opacity$=0$.

However, we noticed that using the edge pixels (*i.e.*, neighboring a background pixel) of the foreground object results in unwanted artifacts near the edge of the object as exemplified in Fig. A1. Considering that non-edge pixels of the foreground usually have a opacity value of $>$0.99, we additionally discard the edge pixels from the cross-view correspondence loss computation by performing 2D average pooling on the opacity map, and disregarding the pixels with opacity values less than the threshold value of 0.99. This step is carried out for both the source and target images, where the predicted target pixel should also be within the non-edge foreground pixels of the rendered view.

2. Soft mutual nearest neighbours filtering. After we compute the 4D correlation map between adjacently rendered views as explained in Sec. 4.3 of the main paper, we perform a soft mutual nearest neighbour filtering as proposed in NCNet [26] to facilitate the reciprocity constraint on matches.

For self-containedness, we provide the details of this approach in the following. Given a correlation map $C^{(i)}$, we perform a soft mutual nearest neighbour filtering $M(\cdot)$ to yield a refined feature map $\hat{\mathcal{C}}^{(i)} = M(\mathcal{C}^{(i)})$, where $\hat{\mathcal{C}}^{(i)}(p,q,r,s) = r^{(i)}_{pqrs} r^{(i+N)}_{pqrs} \mathcal{C}(p,q,r,s)$. $r^{(i)}_{pqrs}$ and $r^{(i+N)}_{pqrs}$ are ratios of the score of the particular match $\mathcal{C}^{(i)}(p,q,r,s)$ with the best scores along each pair of dimensions corresponding to the i-th and $(i+N)$th view respectively:

$$r^{(i)}_{pqrs} = \frac{\mathcal{C}^{(i)}(p,q,r,s)}{\max_{ab}\mathcal{C}^{(i)}(a,b,r,s)}, r^{(i+N)}_{pqrs} = \frac{\mathcal{C}^{(i)}(p,q,r,s)}{\max_{cd}\mathcal{C}^{(i)}(p,q,c,d)}. \tag{10}$$

3. 4D smoothing. We additionally perform a 4D smoothing operation on the correlation map. This not only intends to smooth the 4D space of image correspondences, but also aims at disambiguating correspondences with the help of neighbouring matches. This motivation is inspired by the NCNet [26] as well; we can assume correct matches to have a coherent set of supporting them in the 4D space, especially when the two rendered views depict the *same* object but just from different azimuthal viewpoints. Our 4D smoothing acts as a *soft* spatial consensus constraint (4D convolution with $1/k^3$ uniform weight for each kernel position, instead of learnable weights) to avoid errors on ambiguous or textureless matches.

4. Multi-layer features. We aim to leverage the multiple features that can be obtained along the varying depths of the upsampling layers of the diffusion U-Net. It has been empirically demonstrated in existing work on image correspondences [6, 14, 15] that it is beneficial to leverage features from multiple layers, to exploit both the semantics/context and local patterns/geometries that are encoded in different layers. Among the 12 upsampling layers of the diffusion U-Net in the multi-view diffusion model proposed by MVDream [28], we extract features from the 6th and 9th upsampling layers. We tried with various other combinations, but using these two layers qualitatively gave satisfactory results with reasonable computation overhead.

5. Epipolar constraint. For each of the rendered views from NeRF, we know the ground-truth camera parameters, specifically the extrinsic and intrinsic parameters. Using these, we can accurately determine the epipolar line on the target view corresponding to any pixel on the source view. The epipolar constraint states that the true corresponding point of a point from the source view must lie on the epipolar line. Adhering to this constraint, we project all predicted target points to their respective epipolar lines. To discard obviously wrong correspondences, we calculate the projection distance to the epipolar line in order to discard any correspondence whose projection distance to the epipolar line is larger than a pixel distance threshold τ_{epi}. We use $\tau_{\text{epi}} = 2$ in our settings.

Out-of-bounds filtering. Finally, we filter out any predicted target points which fall out of the non-edge foreground pixels. Also, in calculating the $\text{corr}_{\text{NeRF}}$ from NeRF reprojections, we filter out any reprojections if they fall out of the image bounds of H', W'.

B. Additional implementation details

In this section, we provide additional implementation details of CorrespondentDream. It is noteworthy that CorrespondentDream was implemented largely on threestudio [4] and MVDream [28], and the majority of settings detailed below can be manipulated on their codebase.

We follow the protocols outlined in MVDream [28] to embed the camera embeddings together with the time embeddings as residuals, by adding them together prior to being input to the diffusion network. To prevent the model from generating 3D models with low quality appearance and style, we further add a few fixed negative prompts during the SDS optimization, *e.g*., ”blurry” or ”low quality”, following MVDream [28].

We sample $t \sim \mathcal{U}(t_{\text{min}}, t_{\text{max}})$ where t_{min} anneals from 0.98 to 0.02, and t_{max} anneals from 0.98 to 0.5, both in a linear manner for 9600 iterations. This is the same as MVDream [28], except that the annealing iterations were increased in proportion to the increased number of total iterations. For the first 6000 iterations, we render views from the NeRF at image dimensions of 32×32 at batch size of 8 (*i.e*., 2 sets of 4 views, total of 16 views rendered for cross-view correspondence loss). After the 6000th iteration, we render views from the NeRF at image dimensions of 128×128 (*i.e*., 1 set of 4 views, total of 8 views rendered for cross-view correspondence loss). Beginning the NeRF optimization with rendered views at a lower resolution drastically reduces VRAM usage as empty space is pruned in early training. We set $\omega = 50$ for our class-free guidance. We also devise a class-free guidance scheduling scheme for improved 3D generation quality in the next section (Appendix C). We use a rescale factor of 0.5 for the CFG rescale trick [19]. We turn on soft shading [18] and point lighting [25] to regularize the geometry. We also use the modified version of the orientation loss [34] as in DreamFusion [25], to penalize normal vectors facing backwards away from the camera for the first 6000 iterations. This orientation loss is weighted with a weight that scales linearly from 10 to 1000 until the 6000th iteration. The background is replaced with 50% chance to force the separation between the foreground and the background during the NeRF optimization.

C. Class-free guidance scheduling

Note that we are using a default CFG value of $\omega = 50$ otherwise mentioned in our qualitative visualizations. Through various experiments, we noticed that the value of class-free guidance (CFG) ω has a dramatic effect on the level of details and smoothness of the rendered 3D object. Specifically, we observed that using a large value for ω 1) results in a larger 3D object, 2) results in a higher level of details in the 3D object, but 3) has a larger risk of 3D infidelities. On the other hand, we noticed that using a low value for ω 1) results in a smaller 3D object, 2) results in a smoother surface of 3D objects, but 3) holds a much lower level of details (overly smoothed) in the 3D object. To this end, we experiment with various CFG scheduling schemes at an aim to devise a scheme to maximize the benefits of both high and low CFG values, while minimizing their downsides. Fig. A5 visualizes the results of different scheduling schemes we tried. $\text{CFG}_{10\rightarrow50}$ denotes using $\omega = 10$ for first half, and $\omega = 50$ for the latter half of the 3D generation process. $\text{CFG}_{50\rightarrow10}$ denotes using $\omega = 50$ for first half, and $\omega = 10$ for the latter half of the 3D generation process. $\text{CFG}_{50\rightarrow10\rightarrow50}$ denotes using $\omega = 50$ for first third, and $\omega = 10$ for the second third, and $\omega = 50$ again for the last third of the 3D generation process.

It can be seen that beginning at $\omega = 10$ results in a smaller object, and ending with $\omega = 10$ results in oversmoothed surfaces. While beginning at $\omega = 50$ results in a larger object in comparison, ending with $\omega = 50$ seems to end up with more severe cases of 3D infidelities. The qualitative results show that $\text{CFG}_{50\rightarrow10\rightarrow50}$ exhibits a larger 3D object size, and an appropriate trade-off between the smoothness and detail of the generated 3D shape. Nonetheless, CFG scheduling alone is insufficient to alleviate the 3D infidelities - and it shows that incorporating CorrespondentDream together with $\text{CFG}_{50\rightarrow10\rightarrow50}$ exhibits the best qualitative results overall.

D. Correspondence visualization

CorrespondentDream leverages the annotation-free cross-view correspondences to guide the erroneous NeRF depths, consequently correcting the 3D infidelities. We provide the visualizations of the correspondences in Figs A6 to A8. Specifically, the 3rd and 4th columns depicts the disparity between the cross-view correspondences ($\text{corr}_{\text{diff}}$) and the NeRF correspondences from reprojection ($\text{corr}_{\text{NeRF}}$), where brighter regions have higher disparities, *i.e.*, higher chances of infidelities. The 5th and 6th columns illustrate the cross-view correspondences with the top 20% disparity values. While we do not have the ground-truth correspondences to quantitatively evaluate the quality of the cross-view correspondences, it can be visually seen that the cross-view correspondences are coherent to human perception.

E. Correspondence as geometry pre-/post-processing

In our current scheme, we are supervising NeRF as our 3D output representation in an *alternating* manner using the cross-view correspondence loss $\mathcal{L}_{\text{corr}}$ and $\mathcal{L}_{\text{SDS}}$, in the *midst* of the NeRF optimization process.

In this section, we provide comparative qualitative results of different schemes of leveraging the cross-view correspondences, (1) using only $\mathcal{L}_{\text{corr}}$ for a fixed number of iterations in the middle of NeRF optimization to fix any errors prior to refining the details of the 3D outputs, and (2) using $\mathcal{L}_{\text{corr}}$ as a post-processing refinement to correct the 3D infidelities after the SDS optimization is completed.

We show the results of this comparative experiment in Fig. A9, where it can be seen that our current scheme yields the best results in comparison to the pre-processing or post-processing schemes. For the suboptimal results when using pre-processing, we conjecture this is because the 3D appearance of the output is premature at earlier stages, and using the cross-view correspondence loss at that stage strongly limits the geometric appearance of the output. This is can be particularly detrimental in the potential presence of any erroneous correspondences, and using the $\mathcal{L}_{\text{corr}}$ alone without $\mathcal{L}_{\text{SDS}}$ may lead to the accumulation of errors.

We also assume this to be the reason behind the failure of using $\mathcal{L}_{\text{corr}}$ solely as a post-processing method. While the accumulated errors can be somewhat alleviated via the remaining SDS-supervised iterations when using the preprocessing scheme, the post-processing scheme has no way to alleviate the accumulated errors from 2,000 iterations of cross-view correspondences.

F. Effect of image resolution on output

As mentioned in Sec. 5, while MVDream [28] finally uses NeRF rendered views at resolutions of 256×256, we use final rendered view resolutions of 128×128. We observed this maintains the 3D output quality and infidelities, while incurring significantly less latency and memory overhead. Therefore, we determined that using rendered view resolutions of 128×128 was sufficient to evaluate the efficacy of CorrespondentDream while using lower computational resources.

In this section, we qualitatively evidence that even with the lower rendered-view resolutions, the overall output quality and infidelities are nearly consistent in Fig. A10.

G. Progressive visualization

In this section, we provide the progressive visualization of how the NeRF is optimized in CorrespondentDream, through 2D rendered views along the course of training. We provide a comparison with MVDream [28] to see how our cross-view correspondence loss shows to correct the 3D infidelities. The visualizations are shown Figs A11 to A14, where it can be seen that CorrespondentDream fixes the 3D infidelities along the 3D optimization process, with the help of cross-view correspondences. This is unlike MVDream [28], where the 3D infidelities remain unresolved.

H. Applicability to other models

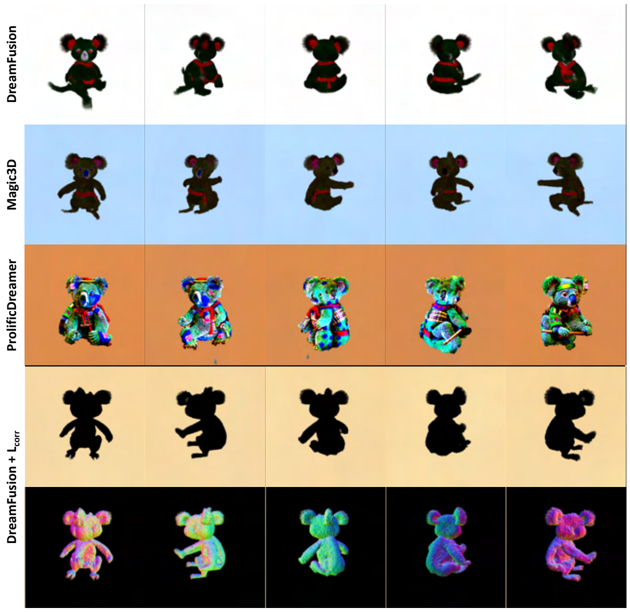

Figure A2. **Results of existing text-to-3D generation methods, and applying $\mathcal{L}_{\text{corr}}$ to DreamFusion [25].** Existing text-to-3D methods suffer severely from 3D inconsistency *e.g.* Janus face problem, which overwhelms the issue of 3D infidelity. The 3D inconsistency makes it challenging to determine accurate cross-view correspondences.

Fig. A2 shows the multi-view rendered results of using the prompt "Samurai koala bear" for 3 different text-to-3D models (DreamFusion, Magic3D, and ProlificDreamer), where they all suffer from the multi-face problem. On the last two rows, we also show that applying $\mathcal{L}_{\text{corr}}$ does not alleviate the multi-face problem, and it is thus hard to determine the 3D fidelity of the output. Thereon, we highlight that while ***our key problem is the presence of 3D infidelities even when the diffusion prior has good 3D consistency***, we also ***rely on good 3D consistency to improve the 3D fidelity***. This is because poor 3D consistency causes the 2D renderings to be incorrect, making it challenging to determine accurate cross-view correspondences. While our method can be integrated with any single/multi-view text/image diffusion priors with *strong 3D consistency*, MVDream was the only such prior at the time of submission.

I. Computation cost analysis

We show the latency and peak GPU vRAM usage for the low-resolution (32×32) and high-resolution (128×128) stages in Tab. A1. Compute requirements can differ by text prompt; here we used "*A zoomed out DSLR photo of a pug made out of modeling clay*". Specifically, the memory usage of $\mathcal{L}_{\text{corr}}$ is similar to $\mathcal{L}_{\text{SDS}}$ despite having to render twice the number of views. However, the latency for $\mathcal{L}_{\text{corr}}$ is higher, as we compute a 4D correlation tensor, and also perform pre-/post- processing to obtain reliable correspondences. We show in Fig. A3 that $\mathcal{L}_{\text{SDS}}$ with increased number of views is still insufficient to alleviate the 3D infidelities, evidencing the efficacy of our method.

	Memory (GB)	Latency (ms)
8-view low-res $\mathcal{L}_{\text{SDS}}$	13.6	197
2×8-view low-res $\mathcal{L}_{\text{SDS}}$	22.5	380
2×8-view low-res $\mathcal{L}_{\text{corr}}$	14.1	485
4-view high-res $\mathcal{L}_{\text{SDS}}$	14.5	198
2×4-view high-res $\mathcal{L}_{\text{SDS}}$	23.5	550
2×4-view high-res $\mathcal{L}_{\text{corr}}$	17.8	4700

Table A1. **Latency and peak GPU vRAM usage for low-resolution (32×32) and high-resolution (128×128) stages.** The memory usage of $\mathcal{L}_{\text{corr}}$ is similar to $\mathcal{L}_{\text{SDS}}$ despite rendering twice the number of views. The latency for $\mathcal{L}_{\text{corr}}$ is higher as we compute a 4D correlation tensor and also perform pre-/post- processing.

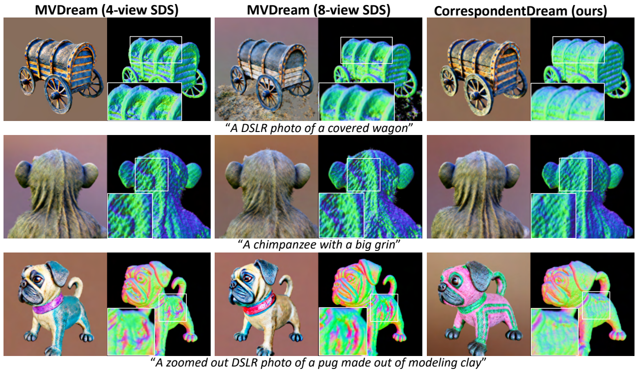
MVDream (4-view SDS)
MVDream (8-view SDS)
CorrespondentDream (ours)
"A DSLR photo of a covered wagon"
"A chimpanzee with a big grin"
"A zoomed out DSLR photo of a pug made out of modeling clay"

Figure A3. **Using $\mathcal{L}_{\text{SDS}}$ only but with double the rendered views.** Using $\mathcal{L}_{\text{SDS}}$ alone is insufficient to solve the 3D infidelities even with double the number of rendered views as in CorrespondentDream.

J. Why not use off-the-shelf matchers?

The capabilities of off-the-shelf matchers rely heavily on the dataset they were trained on, which is problematic where the domain of generated 3D object depends on the text prompt. We visualize the results of warping our i) low-resolution intermediate renderings and ii) high-resolution final rendering using PDCNet [32][5] predictions in Appendix J. Note that PDCNet was the off-the-shelf image matcher which was used in SPARF [33]. It can be seen that PDCNet fails to find high-confidence correspondences

[5] We used the official code and MegaDepth-pretrained weights.

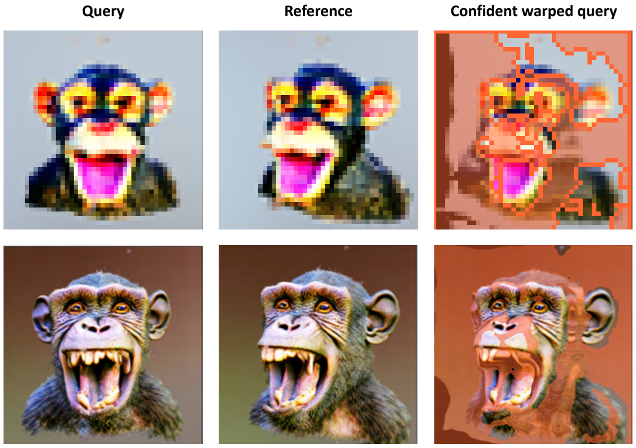

Figure A4. **Visualization of correspondences computed using PDCNet [32].** PDCNet fails to find high-confidence correspondences for most of the foreground regions.

for most of the foreground regions (shown in orange), especially in the low-resolution renderings. Also, off-the-shelf methods incur additional computation; PDCNet incurs approximately 1GB memory usage and 4000ms latency to establish correspondences between an image pair. Using diffusion features eliminates the domain issue without explicit priors or additional compute. However, we believe that a carefully trained matcher could be more effective at handling diffusion features' shortcomings.

K. Example prompts

In this section, we provide some of the prompts which were used to generate the qualitative examples in the main paper and this supplementary material, and the other prompts which were used in our experiments as well in Tab. A2. The prompts were largely borrowed from DreamFusion [25] and MVDream [28].

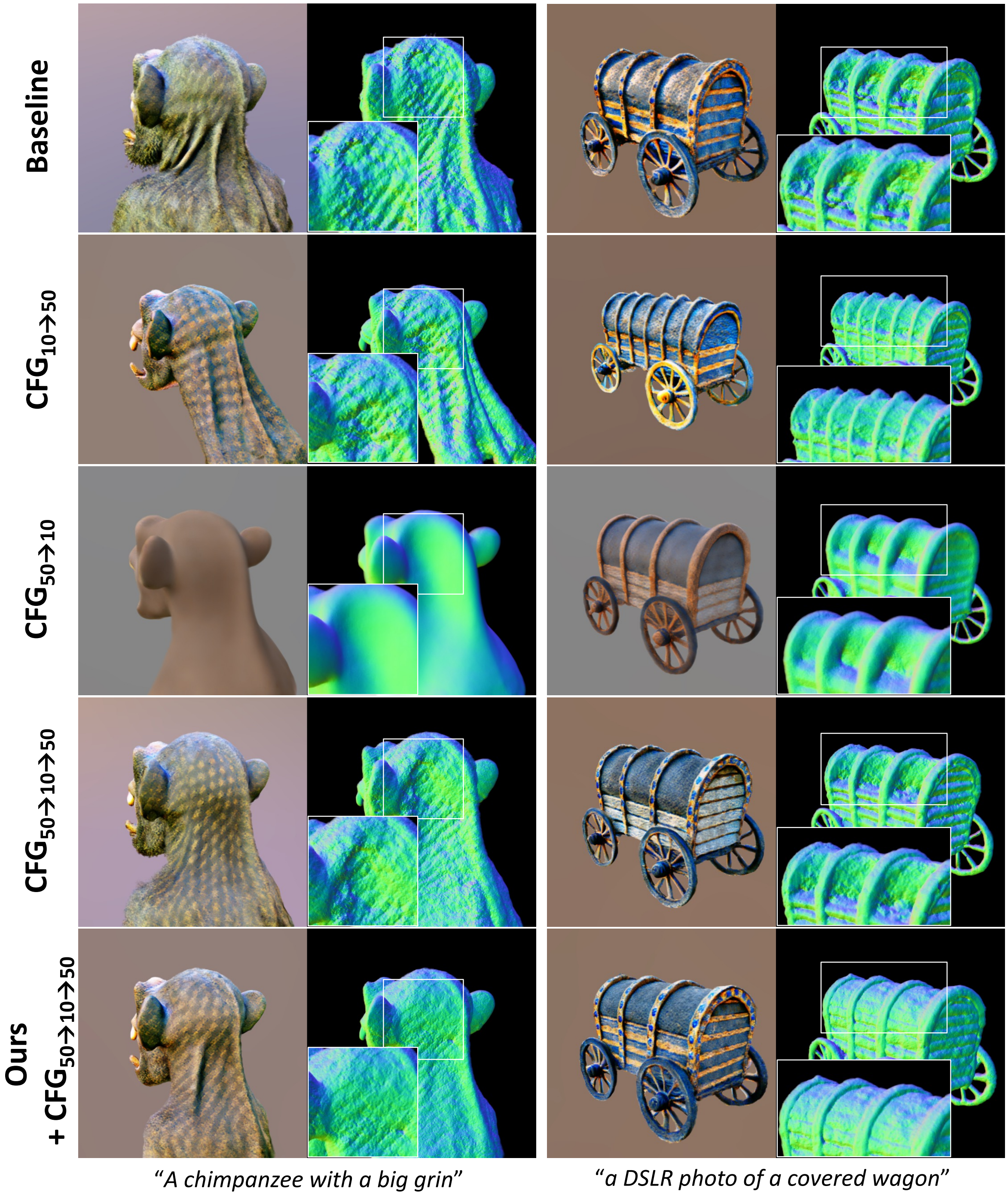
Baseline
CFG$_{10\rightarrow50}$
CFG$_{50\rightarrow10}$
CFG$_{50\rightarrow10\rightarrow50}$
Ours + CFG$_{50\rightarrow10\rightarrow50}$
"A chimpanzee with a big grin"
"a DSLR photo of a covered wagon"

Figure A5. **Visualization of CFG scheduling with and without $\mathcal{L}_{\textbf{corr}}$.** It can be seen that beginning at $\omega = 10$ results in a smaller object, and ending with $\omega = 10$ results in oversmoothed surfaces. While beginning at $\omega = 50$ results in a larger object in comparison, ending with $\omega = 50$ seems to end up with more severe cases of 3D infidelities. CFG$_{50\rightarrow10\rightarrow50}$ exhibits a larger 3D object size, and an appropriate trade-off between the smoothness and detail of the generated 3D shape. Nonetheless, CFG scheduling alone is insufficient to alleviate the 3D infidelities - it shows that incorporating CorrespondentDream together with CFG$_{50\rightarrow10\rightarrow50}$ exhibits the best qualitative results overall.

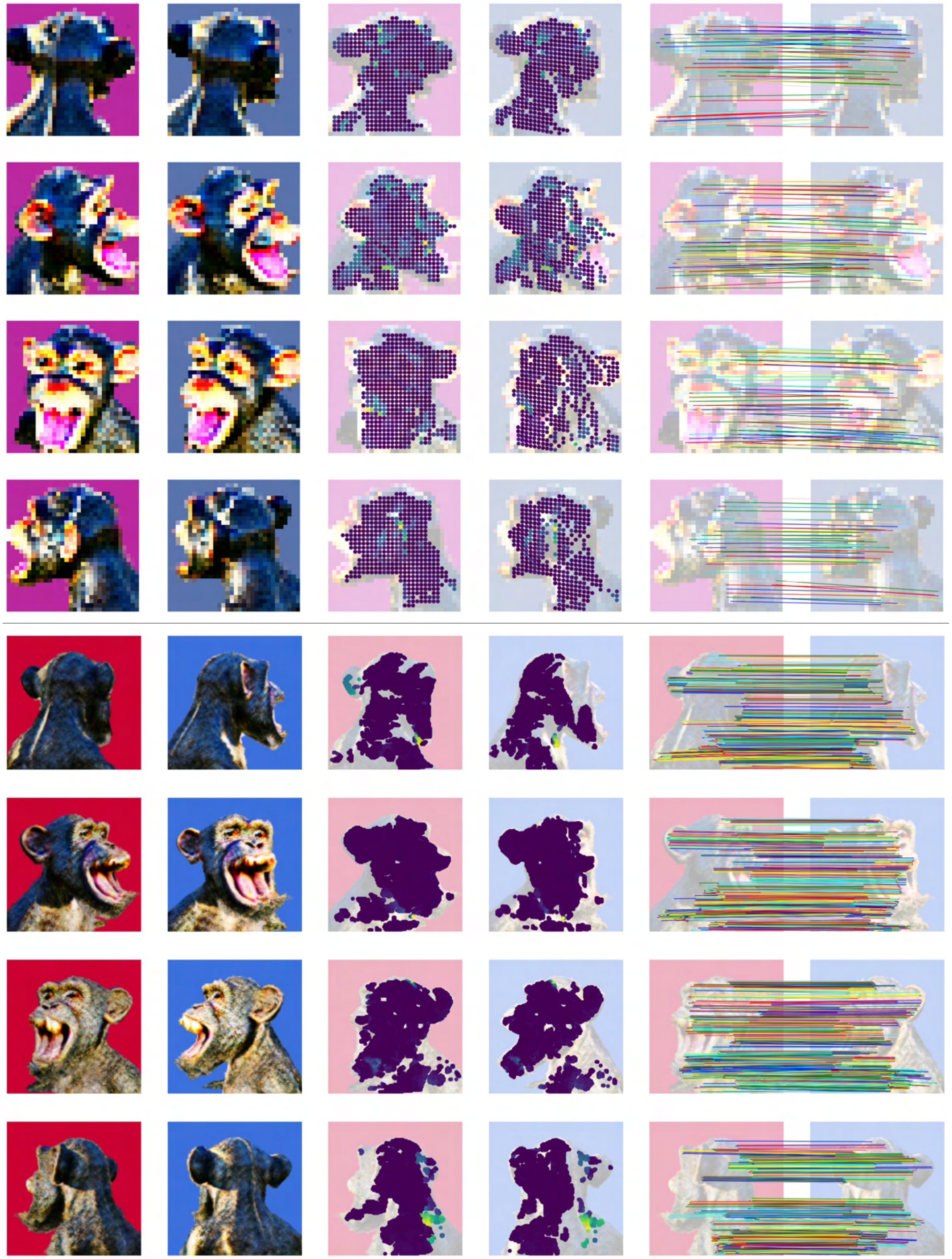

Figure A6. **Correspondence visualization**. Text prompt - ”A chimpanzee with a big grin”. First two columns show rendered views, and the next two columns visualize the difference between the cross-view correspondences and NeRF reprojections, where brighter colours show higher difference. Non-coloured regions show that their correpondences have been filtered out. $\mathrm{corr}_{\mathrm{Diff}}$ correspondences that have the top 20% difference from $\mathrm{corr}_{\mathrm{NeRF}}$ were visualized on the rightmost two columns. The images at top 4 rows were rendered at 32×32, and the lower 4 rows were rendered at 128×128.

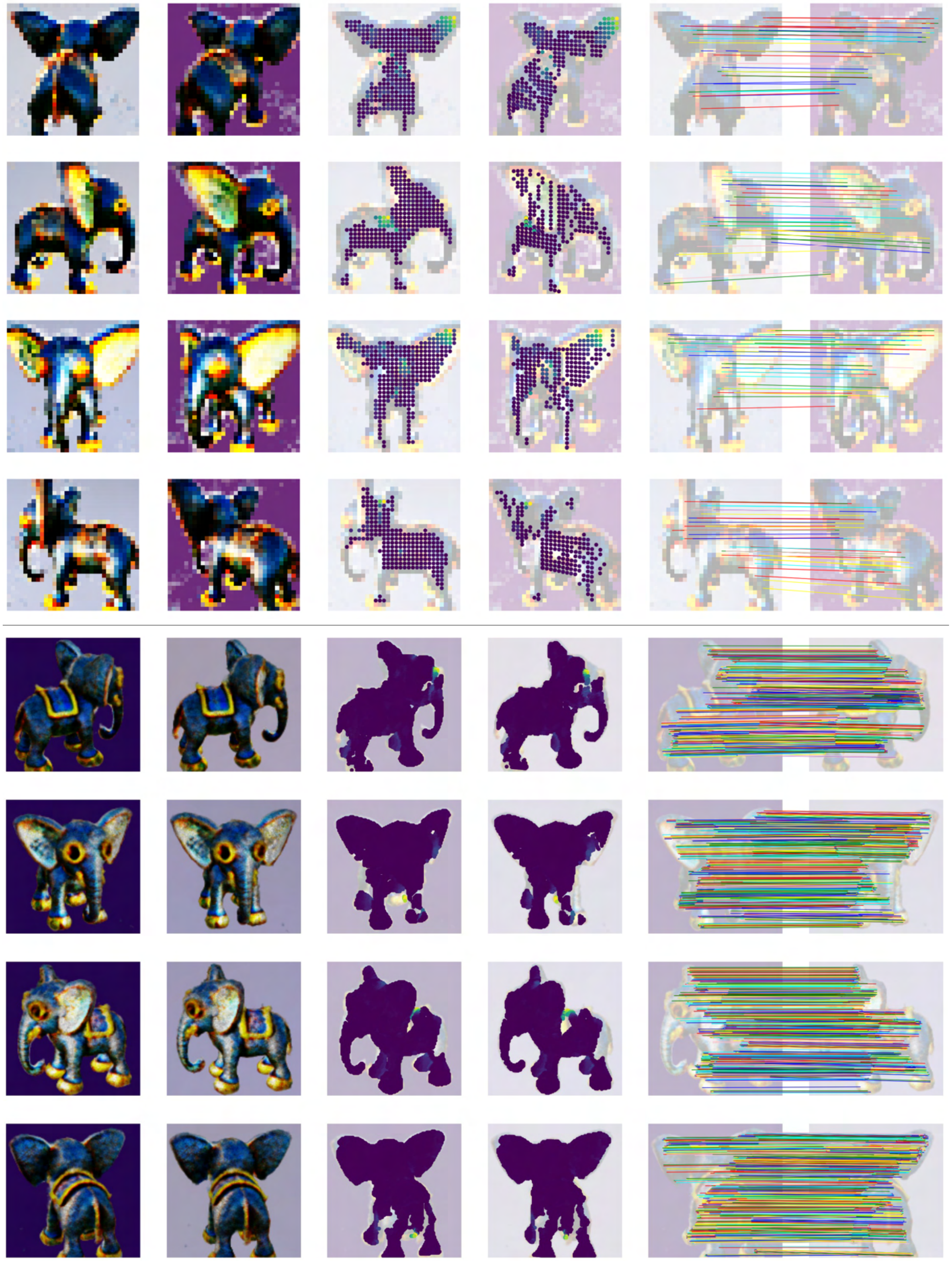

Figure A7. **Correspondence visualization**. Text prompt - ”A cute steampunk elephant”. First two columns show rendered views, and the next two columns visualize the difference between the cross-view correspondences and NeRF reprojections, where brighter colours show higher difference. Non-coloured regions show that their correpondences have been filtered out. $\text{corr}_{\text{Diff}}$ correspondences that have the top 20% difference from $\text{corr}_{\text{NeRF}}$ were visualized on the rightmost two columns. The images at top 4 rows were rendered at 32 × 32, and the lower 4 rows were rendered at 128 × 128.

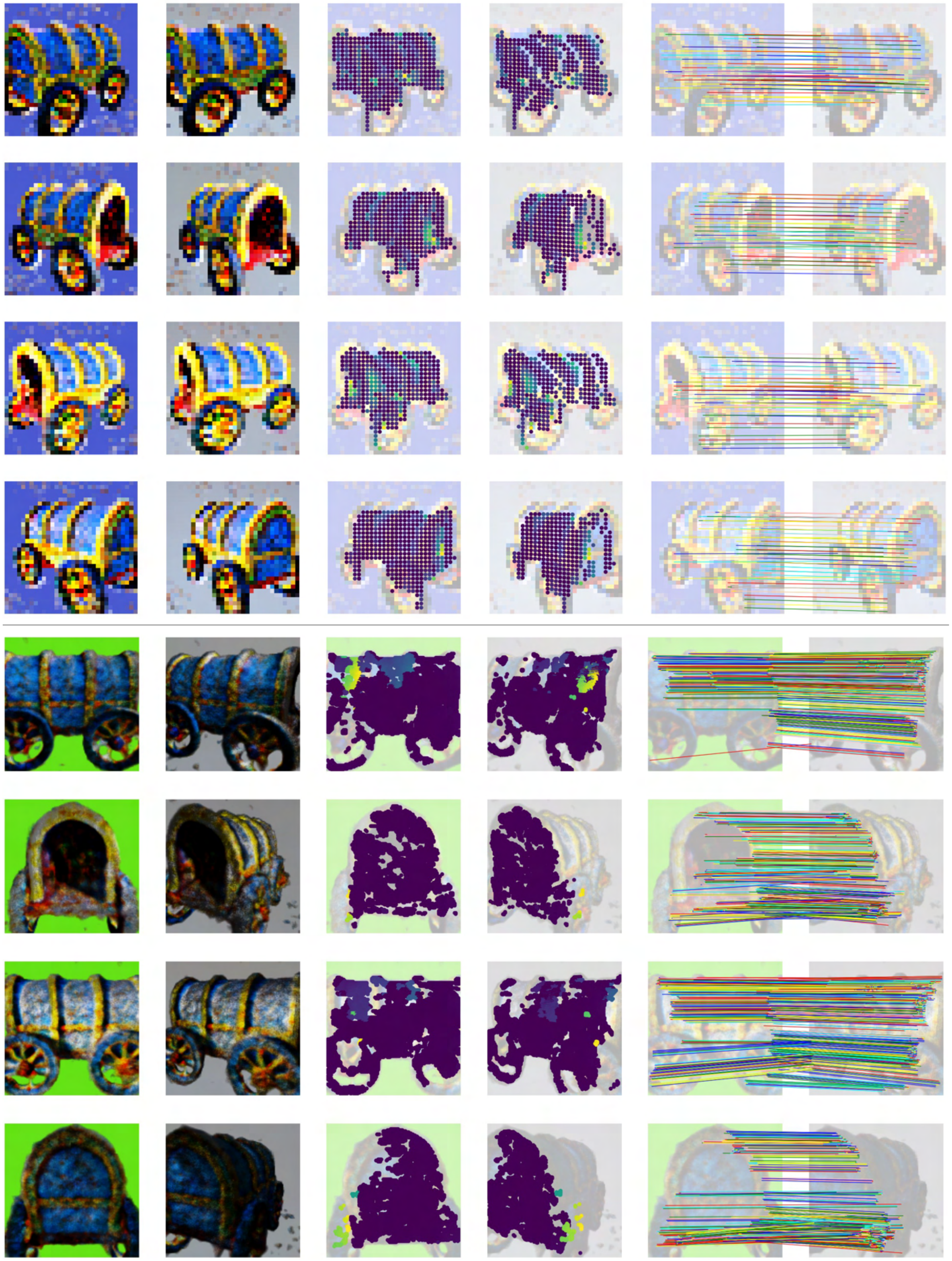

Figure A8. **Correspondence visualization**. Text prompt - ”A DSLR photo of a covered wagon”. First two columns show rendered views, and the next two columns visualize the difference between the cross-view correspondences and NeRF reprojections, where brighter colours show higher difference. Non-coloured regions show that their correpondences have been filtered out. $\text{corr}_{\text{Diff}}$ correspondences that have the top 20% difference from $\text{corr}_{\text{NeRF}}$ were visualized on the rightmost two columns. The images at top 4 rows were rendered at 32×32, and the lower 4 rows were rendered at 128×128.

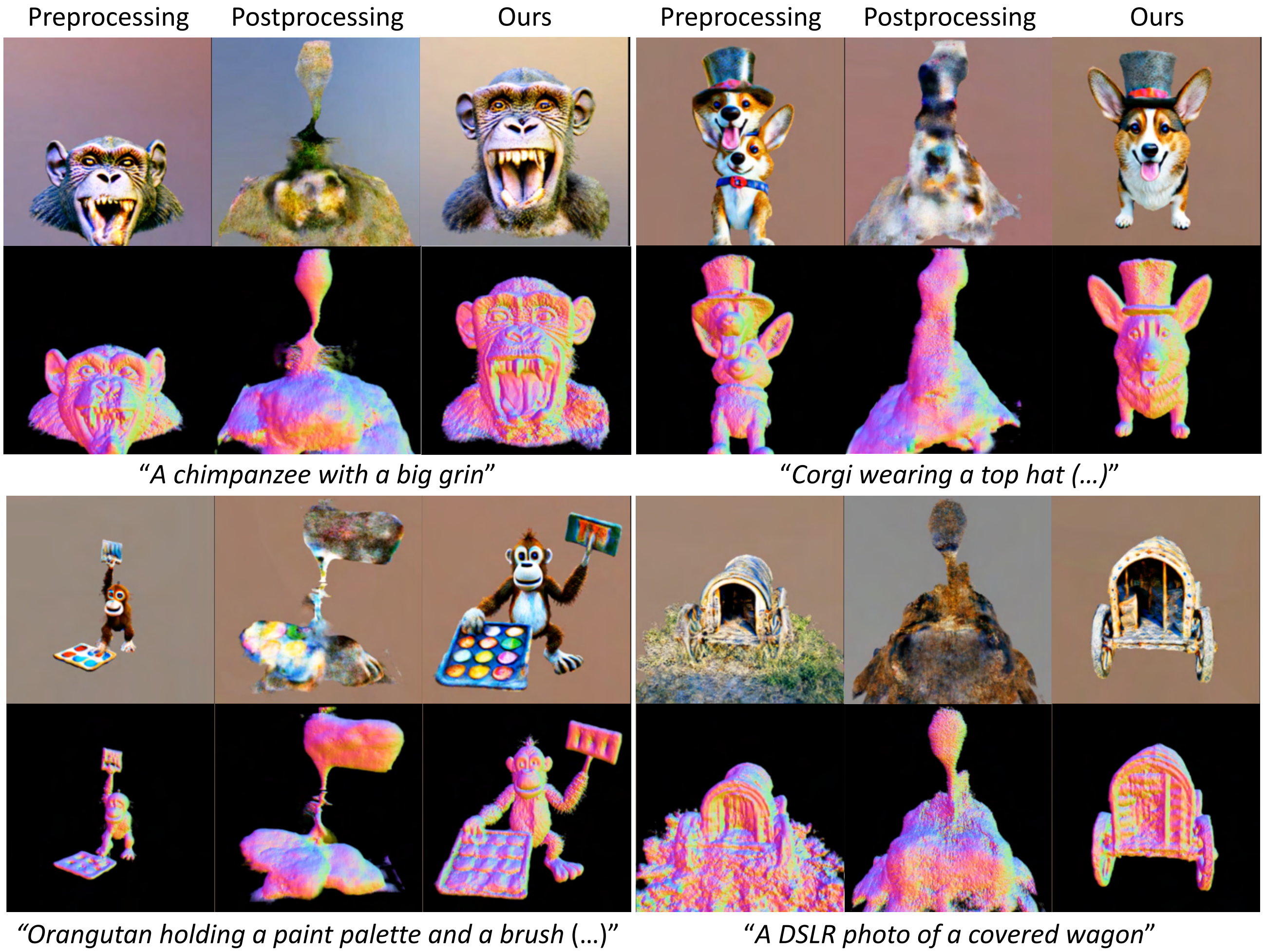
Preprocessing
Postprocessing
Ours
Preprocessing
Postprocessing
Ours
“A chimpanzee with a big grin”
“Corgi wearing a top hat (…)”
“Orangutan holding a paint palette and a brush (…)”
“A DSLR photo of a covered wagon”

Figure A9. **Results when using cross-view correspondence loss as a preprocessing / postprocessing step in NeRF optimization**. Instead of using cross-view correspondence loss as a pre-processing scheme (2,000 $\mathcal{L}_{\text{corr}}$ only iterations after 3,000 iterations, followed by remaining $\mathcal{L}_{\text{SDS}}$ iterations) or post-processing scheme (2,000 L_{corr} only iterations after all $\mathcal{L}_{\text{SDS}}$ iterations have finished), our current scheme of alternating supervision yields the best results.

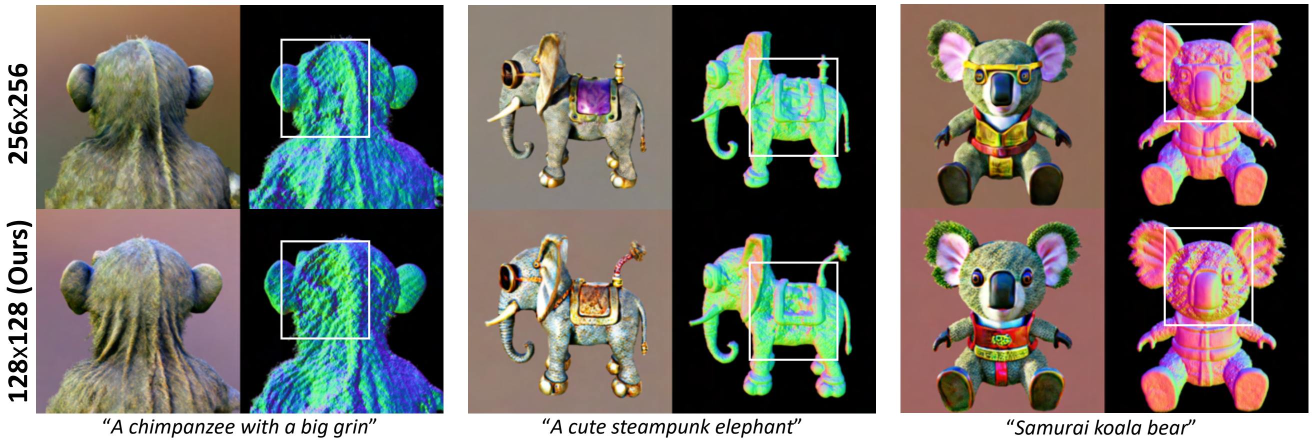

Figure A10. **Comparison of MVDream [28] at different resolutions**. The overall quality of the output, and the 3D infidelities remain even at lower resolutions of 128×128 compared to the original setting of 256×256 of MVDream [28]. We therefore use resolutions of 128×128 in our experiments to quickly validate the efficacy of CorrespondentDream with lower memory and latency overhead.

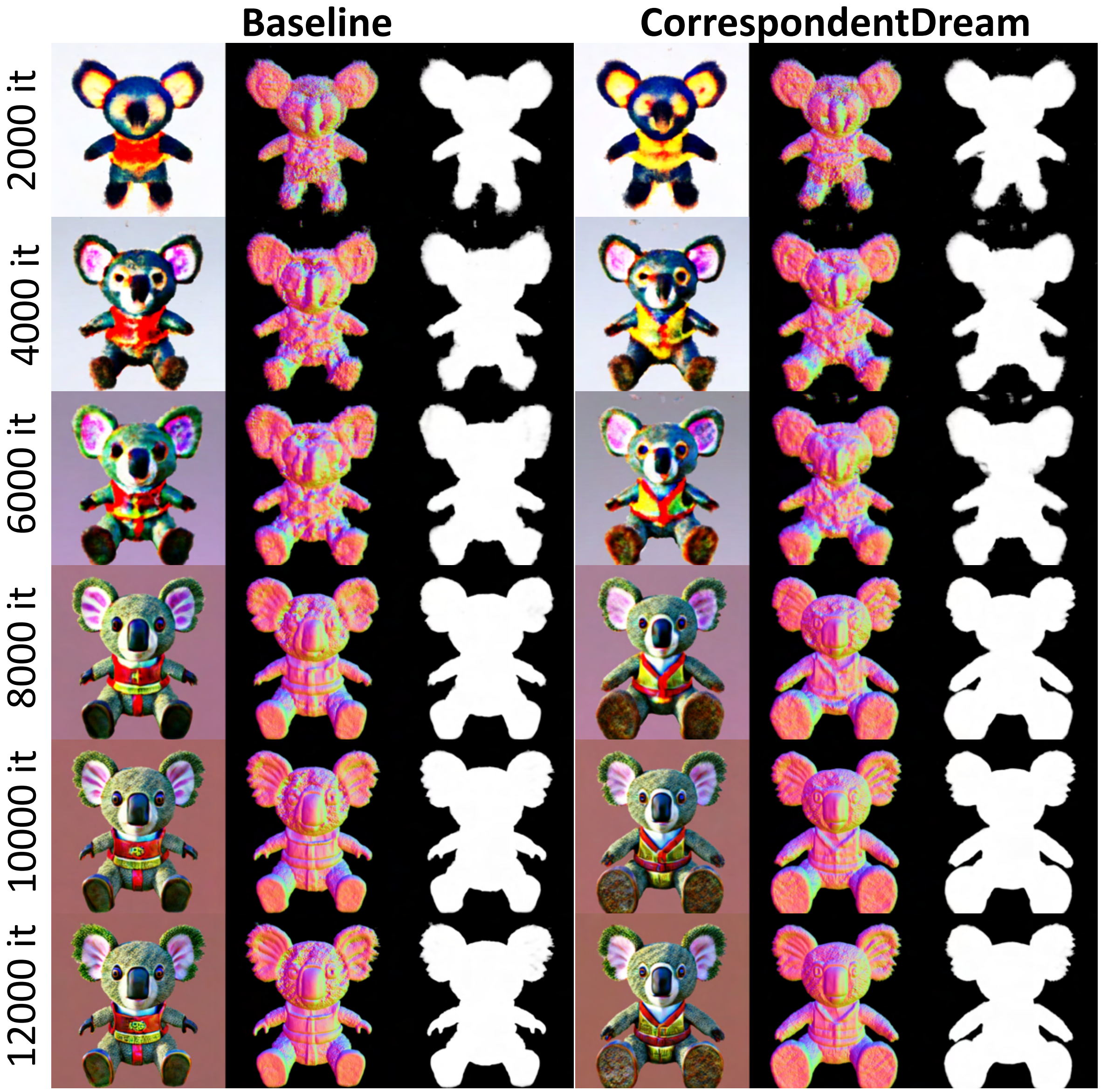

Figure A11. **Visualization of rendered outputs along NeRF optimization**. We visualize the intermediate and final rendered outputs of the baseline (MVDream [28]) and CorrespondentDream for a qualitative comparison. The text prompt used was ”Samurai koala bear”. It can be seen that the 3D infidelities are corrected along the NeRF optimization of CorrespondentDream, whereas the infidelities remain in the baseline.

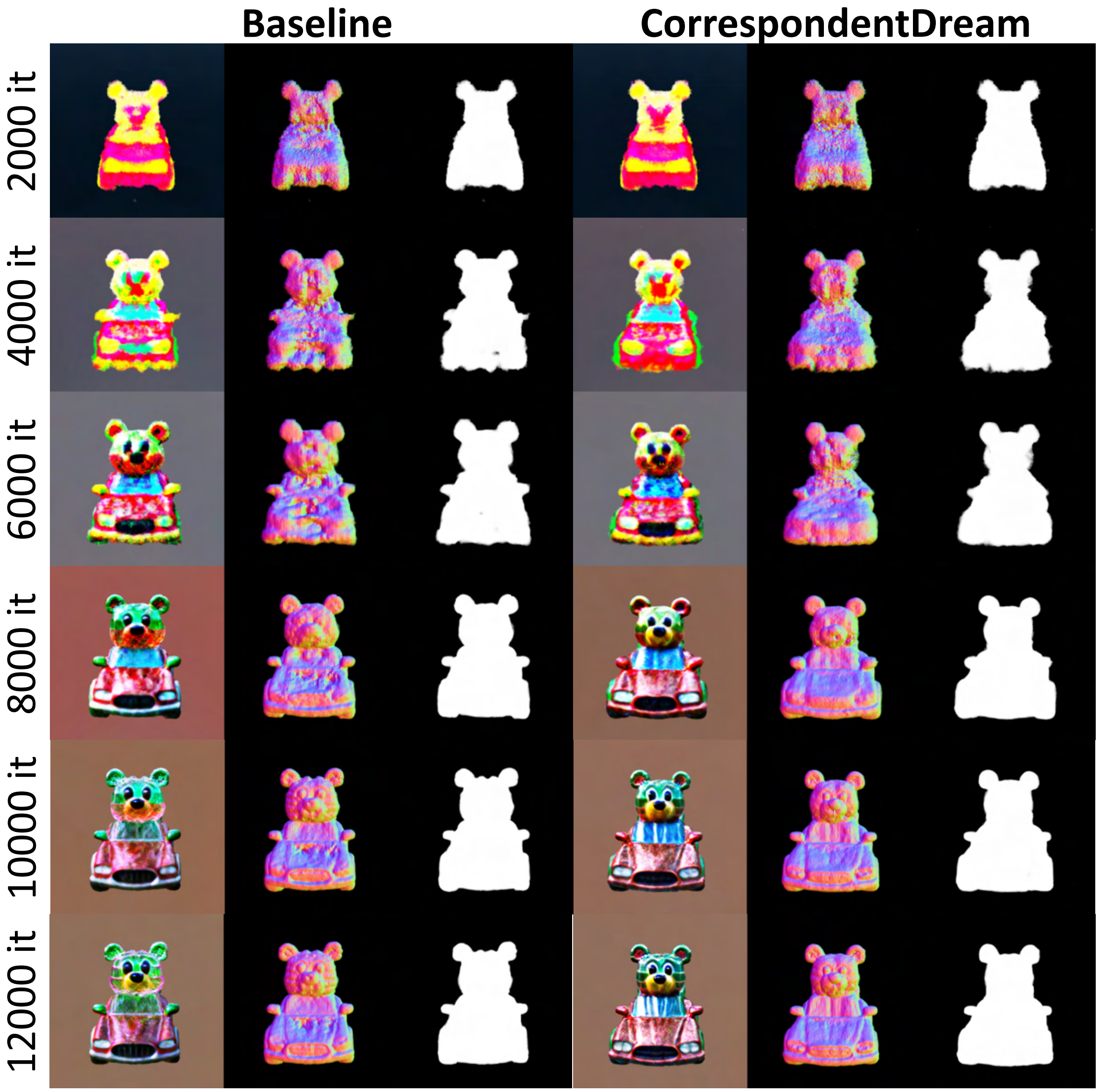

Figure A12. **Visualization of rendered outputs along NeRF optimization**. We visualize the intermediate and final rendered outputs of the baseline (MVDream [28]) and CorrespondentDream for a qualitative comparison. The text prompt used was ”a zoomed out DSLR photo of a gummy bear driving a convertible”. It can be seen that the 3D infidelities are corrected along the NeRF optimization of CorrespondentDream, whereas the infidelities remain in the baseline.

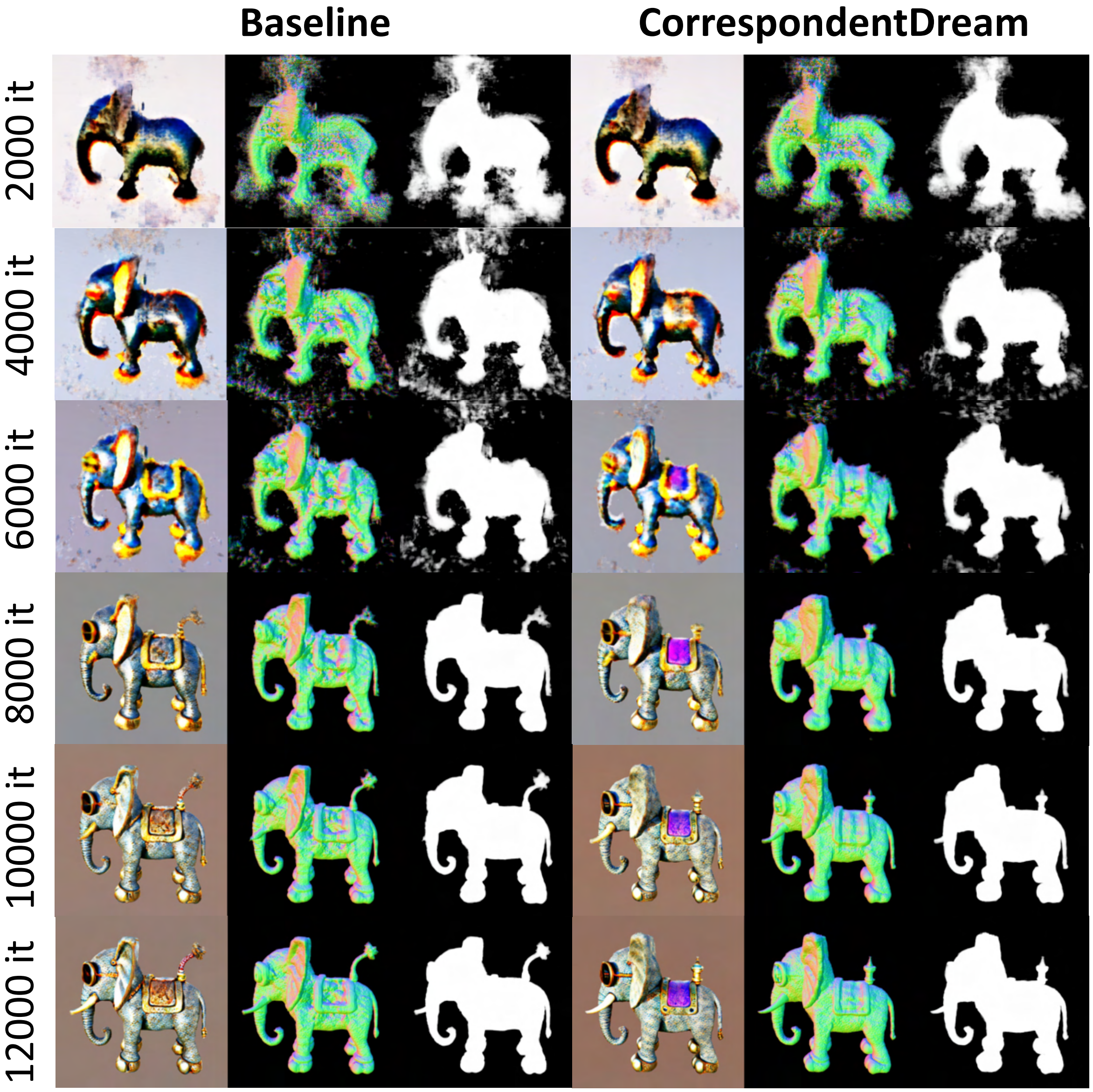

Figure A13. **Visualization of rendered outputs along NeRF optimization**. We visualize the intermediate and final rendered outputs of the baseline (MVDream [28]) and CorrespondentDream for a qualitative comparison. The text prompt used was ”A cute steampunk elephant”. It can be seen that the 3D infidelities are corrected along the NeRF optimization of CorrespondentDream, whereas the infidelities remain in the baseline.

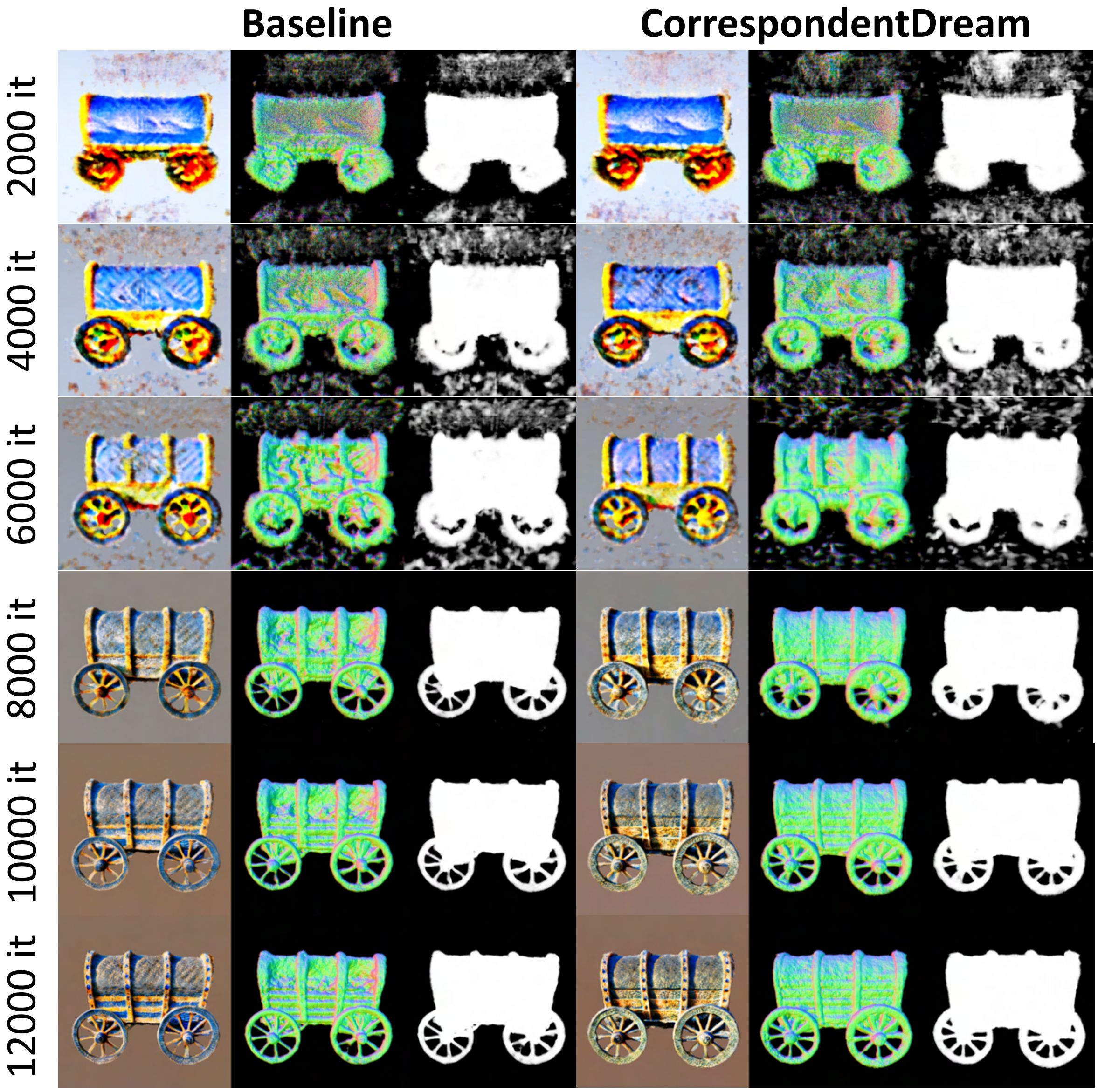

Figure A14. **Visualization of rendered outputs along NeRF optimization**. We visualize the intermediate and final rendered outputs of the baseline (MVDream [28]) and CorrespondentDream for a qualitative comparison. The text prompt used was ”A DSLR photo of a covered wagon”. It can be seen that the 3D infidelities are corrected along the NeRF optimization of CorrespondentDream, whereas the infidelities remain in the baseline.

a bichon frise wearing academic regalia
a capybara wearing a top hat, low poly
a cat with a mullet
a ceramic lion
a chimpanzee with a big grin
a cute steampunk elephant
a DSLR photo of a bear dressed in medieval armor
a DSLR photo of a beautiful violin sitting flat on a table
a DSLR photo of a corgi lying on its back with its tongue rolling out
a DSLR photo of a covered wagon
a DSLR photo of a mug of hot chocolate with whipped cream and marshmallows
a DSLR photo of an iguana holding a balloon
a DSLR photo of a pomeranian dog
a DSLR photo of a porcelain dragon
a DSLR photo of a puffin standing on a rock
a DSLR photo of a pug made out of metal
a DSLR photo of a turtle standing on its hind legs, wearing a top hat and holding a cane
a DSLR photo of a very cool and trendy pair of sneakers, studio lighting
a DSLR photo of a vintage record player
a DSLR photo of cat wearing virtual reality headset in renaissance oil painting high detail caravaggio
An anthropomorphic tomato eating another tomato
an orangutan holding a paint palette in one hand and a paintbrush in the other
a wide angle DSLR photo of a colorful rooster
a yellow schoolbus
a zoomed out DSLR photo of a baby dragon
a zoomed out DSLR photo of a colorful camping tent in a patch of grass
a zoomed out DSLR photo of a corgi wearing a top hat
a zoomed out DSLR photo of a dachsund wearing a boater hat
a zoomed out DSLR photo of a gummy bear driving a convertible
a zoomed out DSLR photo of a hippo made out of chocolate
a zoomed out DSLR photo of an origami bulldozer sitting on the ground
a zoomed out DSLR photo of a pug made out of modeling clay
a zoomed out DSLR photo of a wizard raccoon casting a spell
a zoomed out DSLR photo of a yorkie dog dressed as a maid
an astronaut riding a horse
Samurai koala bear
a DSLR photo of an eggshell broken in two with an adorable chick standing next to it
Darth Vader helmet, highly detailed
Pikachu with hat
A product photo of a toy tank
a boy in mohawk hairstyle, head only, 4K, HD, raw
Wall-E, cute, render, super detailed, best quality, 4K, HD
slayer, assassin with sword, portrait, game, unreal, 4K, HD
an alien monster that looks like an octopus, game, character, highly detailed, photorealistic, 4K, HD
mushroom boss, cute, arms and legs, big eyes, game, character, render, best quality, super detailed, 4K, HD
pentacle sign, 4k, HD

Table A2. **Example prompts.** These prompts were largely borrowed from DreamFusion [25] and MVDream [28].

References

[1] Eric R Chan, Koki Nagano, Matthew A Chan, Alexander W Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. Genvs: Generative novel view synthesis with 3d-aware diffusion models, 2023. 2

[2] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. *arXiv preprint arXiv:2303.13873*, 2023. 2

[3] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022. 3

[4] Yuan-Chen Guo, Ying-Tian Liu, Ruizhi Shao, Christian Laforte, Vikram Voleti, Guan Luo, Chia-Hao Chen, Zi-Xin Zou, Chen Wang, Yan-Pei Cao, and Song-Hai Zhang. threestudio: A unified framework for 3d content generation. https://github.com/threestudio-project/threestudio, 2023. 5, 2

[5] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009. 5

[6] Eric Hedlin, Gopal Sharma, Shweta Mahajan, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. Unsupervised semantic correspondence using stable diffusion. *arXiv preprint arxiv:2305.15581*, 2023. 2

[7] Paul Henderson, Vagia Tsiminaki, and Christoph H Lampert. Leveraging 2d data to learn textured 3d mesh generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7498–7507, 2020. 2

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[9] Susung Hong, Donghoon Ahn, and Seungryong Kim. Debiasing scores and prompts of 2d diffusion for robust text-to-3d generation. *arXiv preprint arXiv:2303.15413*, 2023. 1

[10] Yukun Huang, Jianan Wang, Yukai Shi, Xianbiao Qi, Zheng-Jun Zha, and Lei Zhang. Dreamtime: An improved optimization strategy for text-to-3d content creation. *arXiv preprint arXiv:2306.12422*, 2023. 2

[11] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5885–5894, 2021. 3

[12] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5846–5854, 2021. 3

[13] Animesh Karnewar, Andrea Vedaldi, David Novotny, and Niloy J Mitra. Holodiffusion: Training a 3d diffusion model using 2d images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18423–18433, 2023. 2

[14] Seungwook Kim, Juhong Min, and Minsu Cho. Transformatcher: Match-to-match attention for semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8697–8707, 2022. 4, 2

[15] Seungwook Kim, Juhong Min, and Minsu Cho. Efficient semantic matching with hypercolumn correlation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 139–148, 2024. 5, 2

[16] Ruilong Li, Hang Gao, Matthew Tancik, and Angjoo Kanazawa. Nerfacc: Efficient sampling accelerates nerfs. *arXiv preprint arXiv:2305.04966*, 2023. 7

[17] Xinghui Li, Jingyi Lu, Kai Han, and Victor Prisacariu. Sd4match: Learning to prompt stable diffusion model for semantic matching. *arXiv preprint arXiv:2310.17569*, 2023. 2, 4, 7

[18] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023. 1, 2, 7

[19] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. *arXiv preprint arXiv:2305.08891*, 2023. 3, 2

[20] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. 3

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 7

[22] Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. In *Advances in Neural Information Processing Systems*, 2023. 2, 4

[23] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 3, 4

[24] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 2

[25] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2022. 1, 2, 3, 5, 7, 4, 15

[26] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. *Advances in neural information processing systems*, 31, 2018. 1, 2

[27] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of*

the IEEE/CVF conference on computer vision and pattern recognition, pages 10684–10695, 2022. 2, 7

[28] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15

[29] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Viewset diffusion:(0-) image-conditioned 3d generative models from 2d data. *arXiv e-prints*, pages arXiv–2306, 2023. 3

[30] Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. *arXiv preprint arXiv:2303.14184*, 2023. 2

[31] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *arXiv preprint arXiv:2306.03881*, 2023. 2, 4, 7

[32] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5714–5724, 2021. 4, 5

[33] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from sparse and noisy poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4190–4200, 2023. 3, 5, 4

[34] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. *CVPR*, 2022. 2

[35] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12619–12629, 2023. 2

[36] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *Advances in Neural Information Processing Systems*, 34:27171–27183, 2021. 1, 2, 3

[37] Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, et al. Rodin: A generative model for sculpting 3d digital avatars using diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4563–4573, 2023. 2

[38] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023. 2

[39] Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence. *arXiv preprint arxiv:2305.15347*, 2023. 2