
Supplementary Material for: Mind-the-Glitch: Visual Correspondence for Detecting Inconsistencies in Subject-Driven Generation

A Additional Qualitative Examples

Figures 1 and 2 presents additional qualitative examples of visual and semantic correspondences computed from the disentangled features produced by our architecture under controlled settings (*i.e.* inpainting). As shown in the figure, semantic features tend to match across semantically similar regions regardless of visual appearance, whereas visual features match only in regions with similar appearance. To enable the localization of inconsistencies, we provide heatmap visualizations of the visual feature matching scores, which highlight areas identified as visually inconsistent. This level of spatial interpretability is not offered by existing metrics, including CLIP, DINO, and VLM-based approaches.

We also provide more qualitative examples for the real settings when evaluating subject-driven image generation approaches in Figures 3 and 4.

B Additional Ablation Analysis

We provide additional ablation experiments that are summarized in Table 1. We evaluate on the test set and we report the correlations between the predicted VSM metric and the oracle as explained in Section 4.2 of the main paper.

Why do we need a semantic aggregation branch?: A natural question is why we train a semantic branch at all, given that we could use pre-computed diffusion features, such as those from layer 6 of the decoder, as semantic features, as done in [8, 9]. However, Table 1 shows that using these zero-shot features without a dedicated semantic aggregation network significantly degrades performance. This performance drop arises because, when the semantic and visual branches are trained jointly using a contrastive objective, the two representations become effectively disentangled and spatially aligned. This alignment is essential for accurately computing the VSM metric, which compares semantic and visual features across different regions of the subject to identify inconsistencies. Without spatial alignment, the metric cannot reliably quantify and localize inconsistent regions.

Using 2 Residual Blocks per Layer: Increasing the number of residual blocks per decoder layer in our aggregation network from 1 to 2 results in a slight improvement in Spearman correlation. However, this gain comes at a significant computational cost, increasing the model size by approximately 25%, from $4.5M$ to $6M$ parameters.

Varying the dimensionality of aggregated features q : Increasing the dimensionality of the aggregated features from 384 to 512 results in a significant drop in performance. This may be due to the introduction of redundant channels, which can introduce noisy features and degrade the representation quality, a phenomenon also observed in [3]. Reducing the dimensionality from 384 to 256 leads to a graceful decline in performance, suggesting that $q = 384$ is an optimal choice.

	0-shot Sem.	2× Res.	$q = 256$	$q = 512$	$\alpha = 20$	$M = 2500$	VSM (Ours)
Pearson	0.267	0.472	0.413	0.275	0.412	0.437	0.448
Spearman	0.310	0.453	0.416	0.247	0.369	0.411	0.582

Table 1: Additional ablation analysis of different hyperparameters and design choices. We report the correlation between the VSM metric under different hyperparameters and the oracle on the test set. *0-shot Sem.* refers to using a zero-shot semantic features from a diffusion backbone based on CleanDIFT [8]. *2× Res.* refers to using 2 residual blocks instead of 1 in our aggregation network. $M = 2500$ refers to training on half the size of the training set.

Higher α for the Visual Branch: As demonstrated in the main paper, setting a lower value of $\alpha = 1$ degrades performance by placing insufficient emphasis on learning visual features. Improved results were observed with $\alpha = 10$, indicating better balance between the semantic and visual branches. However, increasing α further to 20 leads to a performance drop, suggesting that $\alpha = 10$ is the optimal trade-off point.

Fewer Training Samples: To examine whether the full training set of 5,000 samples is sufficient for effective feature disentanglement, we train the model using only half of the data, *i.e.*, 2,500 samples, and we train for the same number of epochs. We observe a performance drop of approximately $\sim 10\%$, suggesting that the model already performs well even with half the training data. This indicates that performance gains diminish as more data is added, implying that a training set of 5,000 samples is sufficient to learn this task effectively.

C Sensitivity Analysis of the VSM Metric

To validate whether the proposed VSM metric remains consistent under variations in pose, lighting, and non-rigid deformations, we evaluate it on the DreamBooth [6] dataset. DreamBooth contains real images of the same subject captured under diverse conditions, allowing us to assess the robustness of VSM across environmental and subject-level variations. For this experiment, we selected eight subjects from DreamBooth: *backpack*, *shiny_sneaker*, *duck_toy*, *wolf_plushie*, *robot_toy*, *rc_car*, *bear_plushie*, and *monster_toy*. We refer readers to the official DreamBooth repository for visual examples of these subjects.

For each subject, we manually selected two images exhibiting significantly different poses and, in most cases, additional variations in lighting and non-rigid deformations. We then annotated corresponding regions and applied inpainting to create controlled inconsistencies, following the same procedure as in Section ?? . The oracle score was computed according to the definition in Section 4.2 of the main paper.

Table 2: Correlation between each evaluated metric and the oracle on the DreamBooth subset.

Metric	Pearson Corr.	Spearman Corr.
CLIP [5]	-0.363	-0.309
DINOv2 [2]	0.453	0.047
VLM (ChatGPT-4o)	0.462	0.185
VSM (Ours)	0.846	0.431

As shown in Table 2, VSM exhibits substantially higher correlation with the oracle compared to other metrics, even under large variations in pose, lighting, and deformation. All metrics achieve slightly higher correlations than those reported on the larger test set in Table 1 of the main paper, which we attribute to the smaller size of this DreamBooth subset (eight samples versus 100 in the original test set).

To further assess robustness, we conducted a sensitivity analysis on a larger set of samples. We applied random augmentations, including changes in brightness, contrast, saturation, and horizontal flipping, to all test images (300 images) and examined how the individual metric scores varied under these perturbations. Figure 5 shows that the VSM metric remains stable across all augmentation

types, with only a few outliers marked in red. This consistency demonstrates that VSM is resilient to moderate photometric and geometric variations, supporting its reliability for evaluating visual consistency in real-world subject-driven generation scenarios.

D Additional Details on Benchmarking Subject-Driven Generation Approaches

To demonstrate the effectiveness of our proposed VSM metric in capturing and localizing inconsistencies in subject-driven image generation approaches, we evaluated several recent methods using existing metrics, including CLIP and DINO image-to-image similarity, as well as the VLM-based evaluation from DreamBench++ [4]. For evaluation, we used our test set, where each method is given only the subject image and prompted to generate a new image of the subject in a different environment.

The following methods are evaluated:

- **Diptych Prompting [7]:** We use the official GitHub implementation¹, which is based on the FLUX-dev backbone. Diptychs are generated using the official template, with original image captions and corresponding target caption pairs. Specifically, we use the prompt: *"A diptych with two side-by-side images of the same subject. On the left, a photo of subject. On the right, replicate this subject exactly but as target prompt."* All hyperparameters, including step size, scheduler, and guidance scale, are kept at their default values. Inference is performed at a resolution of 768×1536 , producing a final output of 768×768 . Each sample takes an average of 92 seconds to generate, with a maximum VRAM usage of 55 GB.
- **DSD Diffusion [1]:** We use the official DSD Diffusion repository² with the FLUX model. Inference is performed with 28 steps, a guidance scale of 3.5 (default), and an input resolution of 512×1024 , producing a final output of 512×512 . The model is conditioned on the subject image along with the description and target prompt. Inference is performed using `bfloat16`, taking an average of 8 seconds per sample and a maximum VRAM usage of 34 GB. DSD’s prompt enhancement with Gemini is not used in order to ensure a fair comparison with other methods.
- **EasyControl [10]:** We ran EasyControl³ on the FLUX model using the *subject* control variant, conditioned on the description, target prompt, and subject image. All adapter weights and step parameters were kept as in the original paper: 25 inference steps and a guidance scale of 3.5, with the exception of the resolution, which was reduced from the default 1024×768 to 768×768 . This method exhibited numerical instabilities when using `float16` or `bfloat16`, resulting in completely black outputs for approximately 16% of samples. This issue persisted regardless of resolution or the presence of safety checkers. To avoid this, we ran EasyControl using full precision. On average, inference takes 94 seconds per sample, with a maximum VRAM usage of 70 GB.

E Additional Dataset Filtration Strategy

In our dataset generation pipeline, we apply additional heuristics beyond the matching skewness and LPIPS thresholds described in Section 3.1 of the main paper. These heuristics aim to mitigate failures in segmenting valid subject parts and ensure reliable training data.

Relative Size Check: We compute the relative size of each part mask R_i with respect to its corresponding object mask O_i using the ratio:

$$r_i = \frac{\sum R_i}{\sum O_i}.$$

¹<https://github.com/chaehunshin/DiptychPrompting>

²<https://github.com/cai-lab/DSD-Diffusion>

³<https://github.com/Xiaojiu-z/EasyControl>

We ensure that r_i falls within the range $[0.05, 0.6]$, meaning the segmented region must occupy between 5% and 60% of the object. Additionally, we compare r_1 and r_2 between the two images in a pair and enforce that their relative size difference is less than 0.1.

Relative Aspect Ratio Check: For each part mask R_i , we compute its horizontal and vertical aspect ratios relative to the object mask O_i , defined as the ratios between the width and height of the part and those of the full object. We then compare these ratios between the two images to ensure consistency, requiring that the differences in both horizontal and vertical ratios are within 20%.

F VLM Evaluation Prompts

We follow the same setup as DreamBench++ [4] and we use the GPT-4o model with 0 temperature for deterministic output. We modify the system and user prompts slightly to provide a score from 0 to 100 to align with other metrics and the user-study. We provide the modified prompts in Figures 6 and 7.

G Failure Cases

We include representative failure cases in Figure 8 to highlight current limitations and guide future research. A major challenge lies in detecting incomplete or corrupted generations, as shown in (a) and (b). In the case of missing object parts, it is often unclear to tell whether a part was not generated or simply occluded. For corrupted outputs, identifying whether the subject itself is degraded remains difficult. Another limitation arises with significant lighting changes, as in (c), where visual features fail to match due to the absence of such variations in training data. Lastly, under- or over-segmentation leads to inaccurate VSM scores, since the metric evaluates less than or more than the intended subject region.

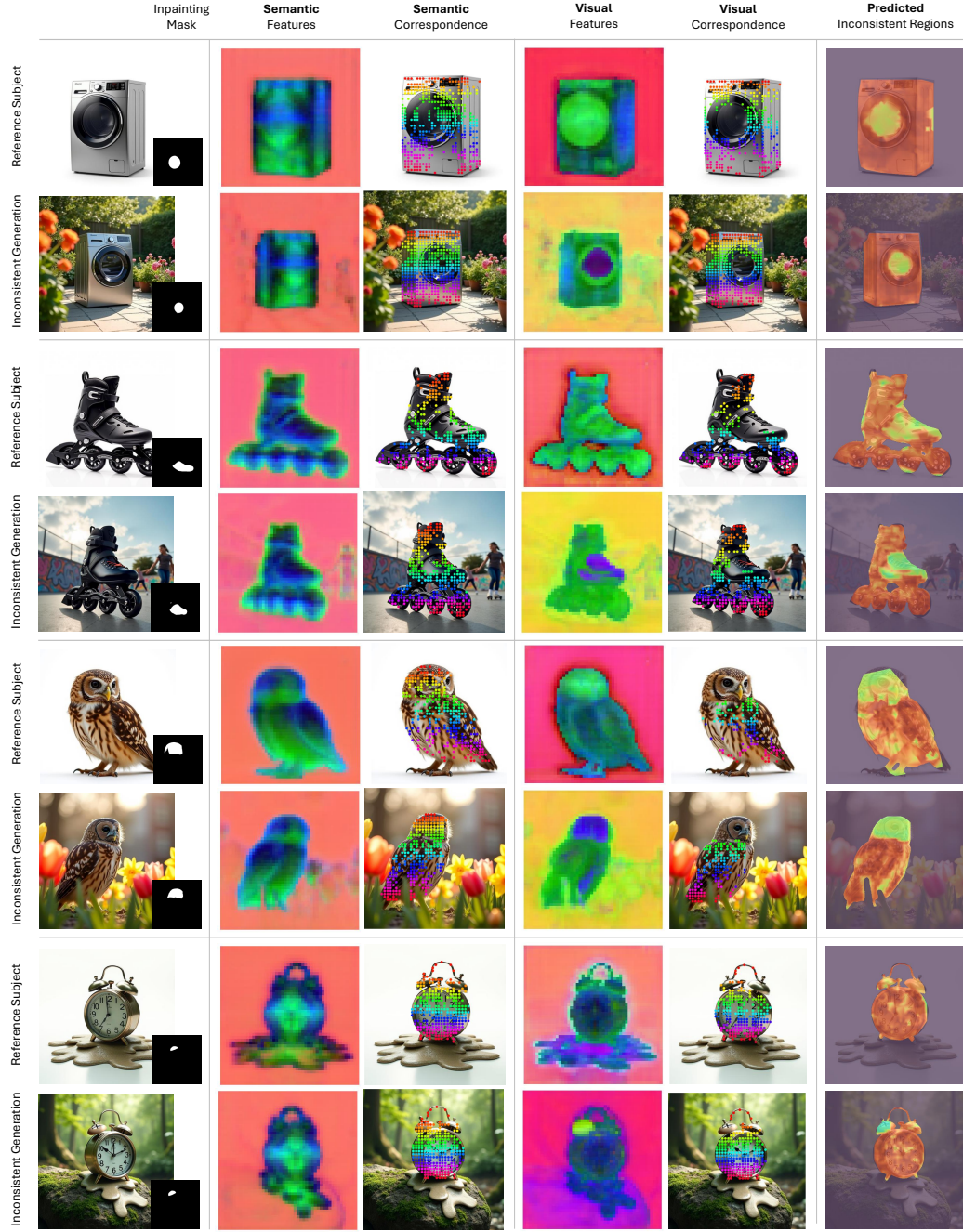


Figure 1: Additional qualitative examples (page 1) of semantic and visual correspondences computed from the disentangled features produced by our proposed architecture. Heatmaps of visual feature matching scores are also shown, highlighting the visually inconsistent regions. **Dark Red** is most consistent and **Yellow** is least consistent.



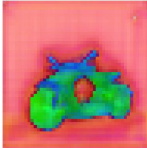




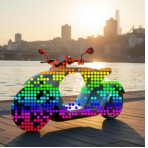







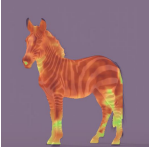



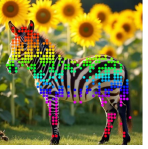
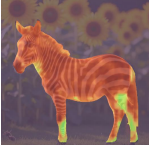
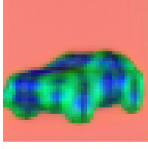



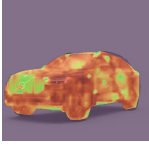
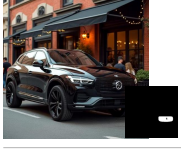
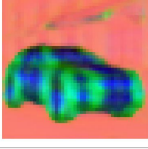
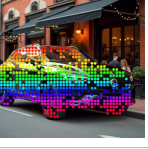

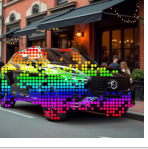





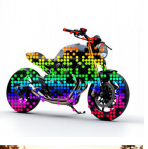
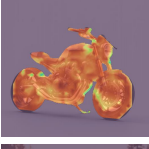
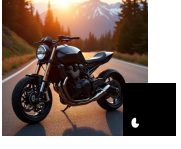

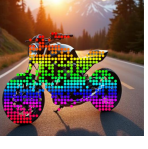
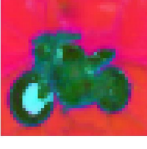
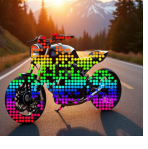

	Inpainting Mask	Semantic Features	Semantic Correspondence	Visual Features	Visual Correspondence	Predicted Inconsistent Regions
Reference Subject						
Inconsistent Generation						
Reference Subject						
Inconsistent Generation						
Reference Subject						
Inconsistent Generation						
Reference Subject						
Inconsistent Generation						

Figure 2: Figure 1 continued.



Figure 3: Additional qualitative examples of evaluating subject-driven image generation approaches using our proposed VSM metric and other existing approaches. Our VSM metric can accurately quantify and localize inconsistency and is more consistent with the oracle. **Dark Red** is most consistent and **Yellow** is least consistent.



Figure 4: Figure 3 continued.

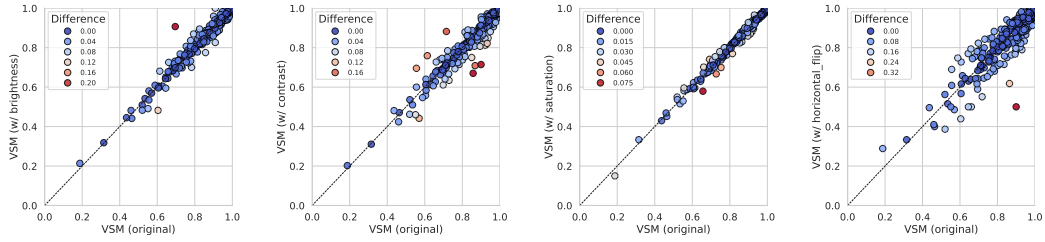


Figure 5: Sensitivity analysis of our VSM metric under varying conditions.

Task Definition

You will be provided with an image generated based on reference image.

As an experienced evaluator, your task is to evaluate the semantic consistency between the subject of the generated image and the reference image, according to the scoring criteria.

Scoring Criteria

It is often compared whether two subjects are consistent based on four basic visual features:

1. Shape: Evaluate whether the main body outline, structure, and proportions of the generated image match those of the reference image. This includes the geometric shape of the main body, clarity of edges, relative sizes, and spatial relationships between various parts composing the main body.
2. Color: Comparing the accuracy and consistency of the main colors generated in the image with those of the reference image. This includes saturation, hue, brightness, and whether the distribution of colors is similar to that of the subject in the reference image.
3. Texture: Focus on the local parts of the RGB image, whether the generated image effectively captures fine details without appearing blurry, and whether it possesses the required realism, clarity, and aesthetic appeal. Please note that unless specifically mentioned in the text prompt, excessive abstraction and formalization of texture are not necessary.
4. Facial Features: If the evaluation is of a person or animal, facial features will greatly affect the judgment of image consistency, and you also need to focus on judging whether the facial area looks very similar visually.

Scoring Range

You need to give a specific integer score based on the comprehensive performance of the visual features above, ranging from 0 to 100:

Here are few examples of how to compute the score.

- Very Poor (0): No resemblance. The generated image's subject has no relation to the reference.
- Poor (25): Minimal resemblance. The subject falls within the same broad category but differs significantly.
- Fair (50): Moderate resemblance. The subject shows likeness to the reference with notable variances.
- Good (75): Strong resemblance. The subject closely matches the reference with only minor discrepancies.
- Excellent (100): Near-identical. The subject of the generated image is virtually indistinguishable from the reference.

Input format

Every time you will receive two images, the first image is a reference image, and the second image is the generated image.

Please carefully review each image of the subject.

Output Format

Score: [Your Score]

You must adhere to the specified output format, which means that only the scores need to be output, excluding your analysis process.

Figure 6: User prompt used for the VLM evaluation using GPT-4o model.

Yes, I understand the task. It involves evaluating the semantic consistency between a reference image and a generated image based on specific criteria. The evaluation focuses on four main aspects: shape, color, texture, and facial features (if applicable). The goal is to determine how closely the generated image matches the reference image in terms of these aspects. The evaluation should result in a specific score ranging from 0 (no resemblance) to 100 (near-identical resemblance).

To evaluate the images, I plan to follow these steps:

1. **Shape**: Assess if the main body outline, structure, and proportions of the generated image are consistent with the reference image. This includes looking at the geometric shape, clarity of edges, relative sizes, and spatial relationships between various parts.
2. **Color**: Compare the main colors in terms of accuracy and consistency, including saturation, hue, brightness, and the distribution of colors.
3. **Texture**: Examine the details in the local parts of the image to see if the generated image captures fine details without appearing blurry and maintains realism, clarity, and aesthetic appeal.
4. **Facial Features**: If the subject includes a person or animal, closely compare facial features to judge visual similarity.

After analyzing these aspects, I will assign a score based on the overall performance of the generated image in relation to the reference image. The score will reflect how similar the generated image is to the reference, strictly adhering to the evaluation criteria provided.

My output format should be Score: [0-100], and I don't need to write out the specific analysis process.

Please provide me with the samples I need to evaluate.

Figure 7: System prompt used for the VLM evaluation using GPT-4o model.

(a) Incomplete generations



(b) Corrupted generations



(c) Excessive Light Changes



(d) Segmentation Failures



Figure 8: Examples of failure cases.

References

- [1] S. Cai, E. Chan, Y. Zhang, L. Guibas, J. Wu, and G. Wetzstein. Diffusion self-distillation for zero-shot customized image generation. In *CVPR*, 2025.
- [2] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.
- [3] T. Darcet, M. Oquab, J. Mairal, and P. Bojanowski. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*, 2024.
- [4] Y. Peng, Y. Cui, H. Tang, Z. Qi, R. Dong, J. Bai, C. Han, Z. Ge, X. Zhang, and S.-T. Xia. Dreambench++: A human-aligned benchmark for personalized image generation. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [5] 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*, pages 8748–8763. PmLR, 2021.
- [6] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023.
- [7] C. Shin, J. Choi, H. Kim, and S. Yoon. Large-scale text-to-image model with inpainting is a zero-shot subject-driven image generator. *arXiv preprint arXiv:2411.15466*, 2024.
- [8] N. Stracke, S. A. Baumann, K. Bauer, F. Fundel, and B. Ommer. Cleandift: Diffusion features without noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.
- [9] L. Tang, M. Jia, Q. Wang, C. P. Phoo, and B. Hariharan. Emergent correspondence from image diffusion. *Advances in Neural Information Processing Systems*, 36:1363–1389, 2023.
- [10] Y. Zhang, Y. Yuan, Y. Song, H. Wang, and J. Liu. Easycontrol: Adding efficient and flexible control for diffusion transformer. *arXiv preprint arXiv:2503.07027*, 2025.