Research Article

VSF: Simple, Efficient, and Effective Negative Guidance in Few-Step Image Generation Models By Value Sign Flip

Wenqi Guo[1], **Shan Du**[1]

1. University of British Columbia, Canada

We introduce Value Sign Flip (VSF), a simple and efficient method for incorporating negative prompt guidance in few-step diffusion and flow-matching image generation models. Unlike existing approaches such as classifier-free guidance (CFG), NASA, and NAG, VSF dynamically suppresses undesired content by flipping the sign of attention values from negative prompts. Our method requires only small computational overhead and integrates effectively with MMDiT-style architectures such as Stable Diffusion 3.5 Turbo, as well as cross-attention-based models like Wan. We validate VSF on challenging datasets with complex prompt pairs and demonstrate superior performance in both static image and video generation tasks. Experimental results show that VSF significantly improves negative prompt adherence compared to prior methods in few-step models, and even CFG in non-few-step models, while maintaining competitive image quality. Code and ComfyUI node are available in https://github.com/weathon/VSF/tree/main.

Wenqi Guo is also affilated to Weathon Software

Corresponding author: Shan Du, shan.du@ubc.ca

 doi.org/10.32388/POY29B

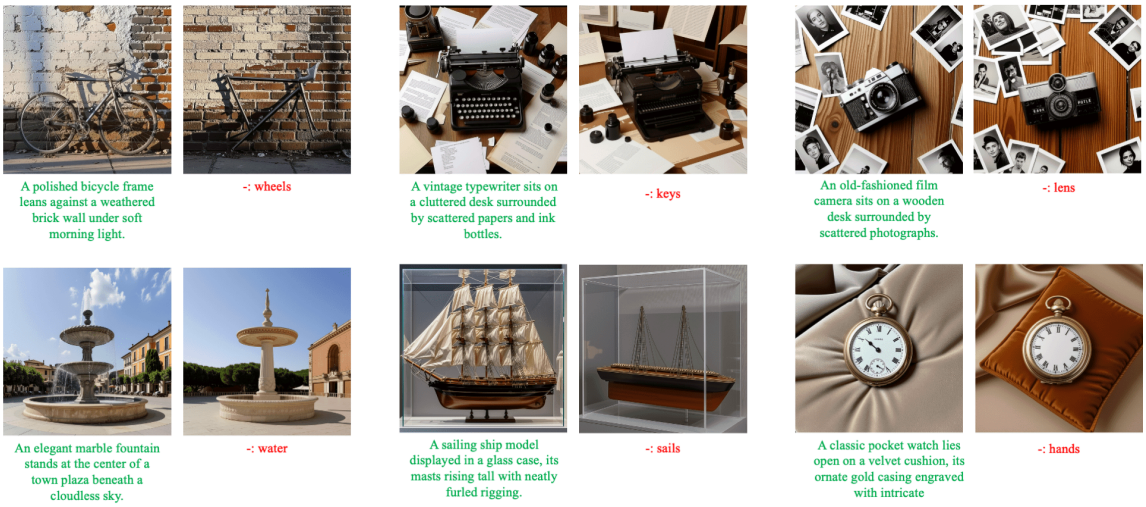


Figure 1. Original image without negative guidance and image generated using our VSF negative guidance on Stable Diffusion 3.5 Large Turbo. The green prompt is the positive prompt, and the red one is the negative prompt. These examples have significant changes as they are removing essential parts of an object.

1. Introduction

Diffusion models (including flow matching models) have demonstrated their ability to produce diverse and high-quality images[1][2][3] and videos[4][5]. However, a longstanding issue persists: negative guidance in image and video generation. Addressing this problem is crucial for improving content control, moderation[6], quality assurance, and reducing biases when generating general concepts[7]. However, vision language models (VLMs) have difficulties interpreting negations[8][9][10][11], rendering prompts containing negations ineffective (e.g., a prompt like “a scientist that is not wearing glasses” will usually generate a scientist with glasses, even more frequently than a plain “a scientist”). Classifier-free guidance (CFG)[12] can address this issue when substituting unconditional generation with negative guidance.

To enhance efficiency in image and video generation, numerous models have been distilled to support inference in just a few steps (1-8 steps), such as Flux Schnell[1], Stable Diffusion 3.5 Large Turbo[3], SDXL Lighting Lin et al.[13], and CaucVid LoRA[5][14]. However, CFG is incompatible with these models. These models are usually distilled and run in a guidance scale of 0 or 1 (depending on frameworks), which means only the positive guidance is used, and there is no extrapolation. When CFG is forcibly applied, the

image will usually be over-saturated when the CFG scale is large enough to remove the unwanted concepts. Additionally, when the step counts are too small, the output shows both positive and negative prompts instead of explicitly avoiding the negative prompt[15] due to divergence between positive and negative guidance signals[7]. An example is shown in Figure 2. Additionally, even if CFG works, it requires two forward passes, one for positive guidance and one for negative guidance, which doubles the run time.

Figure 2. An example of when CFG is forcefully applied to step distilled models, the example is shown using a guidance scale of 2.8 and a step of 4 on SD-3.5-Large-turbo. The positive prompt is about a Canadian winter and a capybara, and the negative prompt is “snow”, we can see that it merges the two concepts unnaturally together and has severe over-saturation artifacts.

To address this, two methods, NASA[15] and NAG[7], have been introduced, employing negative guidance within attention space rather than the output space. NASA is currently limited to cross-attention models, while NAG primarily targets quality control rather than negative prompt avoidance. Both methods calculate positive and negative attentions separately and subtract them using a prefixed scale, resulting in a fixed guidance strength throughout the generation and on different areas on the image. This approach lacks adaptability to various time steps, layers, or image regions, limiting effectiveness in negative prompt adherence[6][16][17].

In this study, we introduce Value Sign Flip (VSF), a method that dynamically adjusts the guidance strength by flipping the sign of negative prompt values during attention. This enables the model to steer away from negative concepts adaptively based on their current presence strength, similar to the approach of Koulischer et al.[16]. VSF has a small computational overhead and, when combined with few-step models, facilitates extremely fast image or video generation.

2. Related Work

2.1. Negation in Vision Language Models

Much previous work has shown that existing vision language models (VLM) struggle to understand negation[18][11][9][8]. In classification tasks, the model cannot correctly understand text with negation in it (e.g. “a dog running” vs “a dog not running” might have very close embeddings, even though they are opposite). This problem has been introduced into text-to-image generation tasks, making it hard for the model to generate images without certain concepts (examples in Figure 1 of Singh et al.[10] and Figure 5 of Park et al.[8]). Thus, classifier-free guidance (CFG) was used to introduce a negative prompt to the image generation process. More details in the next subsection. Several studies have attempted to tackle this issue by employing alternative training strategies, such as incorporating harder samples in the training data designed for negation tasks[18][11][10][9][8]. Some of these methods have shown improvements in image generation tasks. For instance, Park et al.[8] reported gains in Neg Score—measuring whether the model retains the primary subject while correctly omitting the negated object—for both SD-1.4 and SDXL-1.0, by replacing the default CLIP encoder with their NegationCLIP on their dataset, without additional T2I training. Nonetheless, the Neg Score remained below 0.5, indicating limited effectiveness.

These methods generally require re-training the text encoder (usually a CLIP-like model) with contrastive learning, which poses challenges for models that do not use contrastively pre-trained encoders, such as T5[19] in Stable Diffusion 3[3][20] and Flux[1]. Moreover, each model using a different text encoder would require a separate, dedicated adaptation.

2.2. Classifier Free Guidance

Original classifier-free guidance (CFG)[12] generates a conditioned noise prediction and an unconditioned noise prediction. In flow matching[21], the predicted targets are the velocity (u_t) pointing to the image. Thus, the original flow matching CFG prediction can be written as

$$u_t = f(\varnothing, x_{t+1}) + \lambda(f(p^+, x_{t+1}) - f(\varnothing, x_{t-1})), \tag{1}$$

where p^+ is the positive prompt, x_t is the latent at time t (where higher t means more torward the noise distribution), $f(\cdot)$ is the trained model, and λ is the guidance scale. Later, the community finds out that by replacing the unconditional generation with a negative prompt (e.g., description of an unwanted

image), the model will avoid the prompt due to the negative sign. This is the common implementation of a negative prompt. This turns the above equation into

$$u_t = f(p^-, x_{t+1}) + \lambda(f(p^+, x_{t+1}) - f(p^-, x_{t+1})), \tag{2}$$

2.3. Recent Works on Negative Guidance

The studies on negative guidance are very limited (only [17][16][6]). Ban et al.[17] finds that the negative prompts affect the model by delayed effects and neutralization. After the model has generated unwanted contents, the negative guided vector (u_{p^-}) will neutralize the content. They also observed the reverse activation effect, where the negative prompt introduced early in the diffusion processes could actually induce the unwanted concepts. To address this, they proposed applying the negative guidance later in the diffusion process and found it effective.

Schramowski et al.[6] used a very similar idea as CFG to avoid unwanted (NSFW) content. They generate an unsafe vector and purposely avoid it by subtracting it from the predicted noise. They also added a pixel-level guidance scale that depends on the pixel-wise distance between the positive predicted noise and the unwanted noise.

Koulischer et al.[16] used similar ideas of both and proposed a temporal dynamic guidance scale method. They calculate a probability that the generated concept contains negative content and adjust the guidance scale accordingly. However, their adaptive scale only changes throughout the steps and does not adapt to different regions in the image.

2.4. Few-Step Image Generation Models

Traditional diffusion or flow-matching image generation models typically require many inference steps. However, with improved schedulers, this can be reduced to around 20 steps. Recent approaches go further by using step distillation to reduce the number of steps to fewer than 8, or even a single step, as demonstrated in Flex Schnell[1], SDXL Lightning[13], CausVid[5][14], and Stable Diffusion 3.5 Turbo[3]. Since these models are distilled, they generally do not use classifier-free guidance (CFG) during inference; when CFG is forcibly applied, the results are significantly degraded to the point that it is completely unusable[15], see Figure 2 for an example.

2.5. Recent Works on Negative Guidance in Few-Step Models

Recently, two approaches have specifically targeted negative guidance techniques for few-shot models: Negative-Away Steer Attention (NASA)[15] and Normalized Attention Guidance (NAG)[7]. Although they both focused on avoiding unwanted content and improving quality (using a negative prompt describing bad quality), NASA mainly focused on avoiding unwanted content, while NAG focused on improving quality.

The authors of the NASA study found that neither standard CFG nor CFG applied directly to text embeddings yields desirable results in few-step scenarios, particularly in single-step settings. Specifically, the regular CFG independently computes positive and negative guidance signals, preventing the negative guidance from effectively neutralizing unwanted concepts. As a result, the produced images merely appear as a mixture of both positive and negative prompts unnaturally (an average image of the positive prompt generated image and the negative prompt independently generated image) rather than excluding negative prompt elements. Furthermore, the authors noted that applying CFG to text embeddings produces minimal benefits. For detailed examples and further illustration, readers are referred to the original paper introducing NASA (SNOOPI[15]).

The method NASA proposed is to apply the guidance in intermediate states instead of the predicted noise or velocity. Specifically, they calculate a positive attention output Z^{+} and a negative attention output Z^{-}, and they output the final attention Z^{NASA} by subtracting the two with a factor α, as shown in Equation 3. The alpha value is usually between 0 and 1.

$$Z^{NASA} = Z^{+} - \alpha Z^{-} \tag{3}$$

3. Proposed Methods

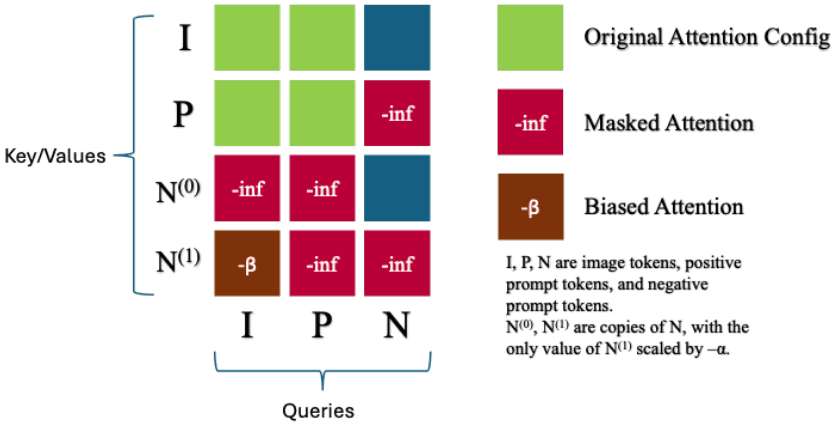


Figure 3. The attention mechanism of our method. We pass in image tokens (I), positive prompt tokens (P), and negative prompt tokens (N) into attention. For key and values, N is duplicated, with values of one copy ($N^{(1)}$ scaled by $-\alpha$. Some areas are masked to avoid interference. An bias $-\beta$ is added to $I \to N^{(1)}$ attention.

Normalized Attention Guidance (NAG) used a similar approach. But instead of subtracting the negative attention map from the positive, it uses a similar extrapolation approach as CFG, as shown in Equation 4. The starting point Z^+ could also be replaced with Z^-; they are equivalent if ϕ is increased by 1.

$$\tilde{Z}^{NAG} = Z^+ + \phi(Z^+ - Z^-) \tag{4}$$

However, to maintain the stability of the attention space, they also applied normalization to $\tilde{Z}^{NAG}$ to limit its norm releative to Z^+ with scale τ, resulting in $\hat{Z}$. Then it used a blending factor α to blend it with the positive attention result, as shown in Equation 5.

$$Z^{NAG} = \alpha\hat{Z} + (1-\alpha)Z^+ \tag{5}$$

The normalization and blending ensure the attention output of the NAG does not drift away from what the model usually sees during training, improving the quality of generated images. However, it also limits the model's following to negative prompt guidance if the constraint is set to be too tight (i.e., high α and low τ).

doi.org/10.32388/POY29B

3.1. Value Sign Flip Adaptive Attention

Our proposed method is similar to NASA. NASA used a fixed value α at all layers and stages for all tokens during the image generation, pushing the content away from the negative output (Z^-) even when the unwanted concept is not present. Previous work[22][16][17][6] has shown that a dynamic or adaptive guidance scale could yield better results. In the NAG[7] future work section, they also hypothesized that token-level modulation may be beneficial. Additionally, in NAG, they applied guidance by extrapolation, which means that to increase the negative guidance scale, you will also need to increase the positive guidance scale. This might make it challenging for cases where positive concepts are closely related to negative concepts (e.g., bike but no wheels) or when the effects needed are contradictory to extrapolation (e.g., when the need to generate camouflaged, undersaturated, or blurry samples is purposely done).

Drawing from[16][6], one possible solution is to steer the latent space away when the model is about to generate, or has already generated, unwanted content. In their approach, a probability-based method is used to determine the appropriate guidance factor. Alternatively, a more intuitive method involves using the model's attention map: when the image attends more to the negative prompt compared to the positive one, it should be steered away stronger accordingly. This can be implemented by simply concatenating the values and keys of the positive and negative prompts, then flipping the sign of the negative prompt values so that when the image attends to the negative prompt, the flipped value of the negative prompt can cancel the unwanted content. Note that the key of the negative prompt is not flipped to keep the original meaning of the unwanted concept to match image patches. The equation of our method in cross attention models, written in the matrix calculation, is shown in Equation 6, where $\oplus$ means matrix concatenation on the sequence length dimension, and σ is the softmax function on the sequence length dimension, Q is the image query tokens, K^+ and K^- are the positive and negative prompt keys, V^+ and V^- are the positive and negative prompt values, and α is the factor controlling the strength of the guidance.

$$Z^{VSF} = \sigma(\frac{Q(K^+ \oplus K^-)^T}{\sqrt{d}})(V^+ \oplus -\alpha V^-) \tag{6}$$

Mathematically, this is equivalent to computing the ratio between the image-to-positive and image-to-negative attention map strength, scaling each attention output using this ratio, and subtracting the negative velocity from the positive one.

This approach gives a dynamic weight for the positive and negative prompts, and it varies for different layers, steps, and tokens.

3.2. Attention Masking and Duplication of Negative Embedding

The above method works well for cross-attention-based methods, where attention only exists between image-to-image in self-attention layers and image-to-text in cross-attention layers. However, it requires modification, including masking and duplication, to work in MMDiT-style models such as SD3.5[3], where all image and text tokens are concatenated into a single sequence before attention.

In the standard MMDiT-style setup without our guidance, the sequence inputs for the attention module are:

$$[\mathbf{I}, \mathbf{P}] \text{and} [\mathbf{I}, \mathbf{N}]$$

If we concatenate all tokens into a single sequence without any modification, we will get:

$$[\mathbf{I}, \mathbf{P}, \mathbf{N}]$$

where $\mathbf{I}$ represents image tokens, $\mathbf{P}$ represents positive prompt tokens, and $\mathbf{N}$ is the negative prompt. During attention, queries, keys, and values are all projected from this combined sequence.

If we apply a sign flip to the negative prompt values by scaling $V_{\mathbf{N}}$ with $-\alpha$ (where V is the value projection, this flipped content affects all attention paths involving $V_{\mathbf{N}}$. That includes not only the intended interaction between image and negative prompt ($\mathbf{I} \to \mathbf{N}$), but also undesired interactions such as positive-to-negative ($\mathbf{P} \to \mathbf{N}$) and negative-to-negative ($\mathbf{N} \to \mathbf{N}$) (which the value will cancel itself). These unintended interactions can distort the behavior of the model since the flipped signal influences more than just the image.

To address this, we introduce a duplication of the negative prompt. One copy remains unflipped and unscaled, denoted $\mathbf{N}^{(0)}$, and the value (and only value) of other is flipped and scaled, denoted $V_{\mathbf{N}^{(1)}} = -\alpha \cdot V_{\mathbf{N}}$. The sequence becomes:

$$[\mathbf{I}, \mathbf{P}, \mathbf{N}^{(0)}, \mathbf{N}^{(1)}]$$

doi.org/10.32388/POY29B

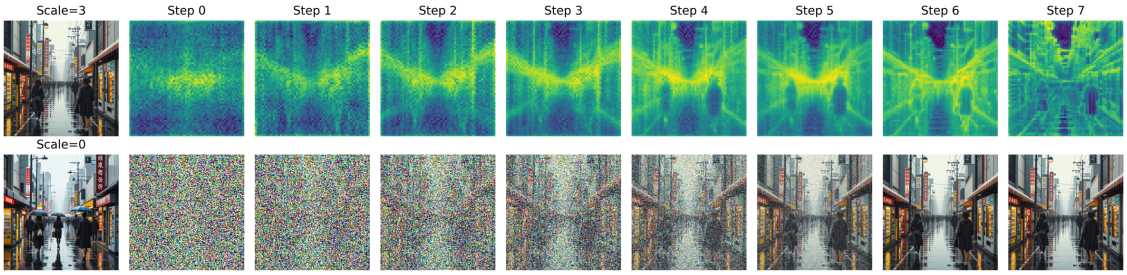


Figure 4. Attention maps and intermediate images during the diffusion process. The leftmost column shows the final generated image (top) and an image generated without applying VSF scaling ($\alpha = 0$, bottom). The top row on the right side displays the unnormalized attention values between image tokens and negative prompt tokens, while the bottom row shows the corresponding intermediate images at each timestep. The negative prompt is "unbrulla."

Queries are sourced from $\mathbf{I}$, $\mathbf{P}$, and $\mathbf{N}^{(0)}$, while keys and values include all four components.

With a similar idea in Wang et al.[23], we apply attention masks to isolate the effect of the flipped negative prompt. Specifically, $\mathbf{N}^{(0)}$ is only allowed to attend to $\mathbf{I}$ and to itself, while $\mathbf{N}^{(1)}$ is only attended to by $\mathbf{I}$. Since $\mathbf{N}^{(1)}$ does not act as a key or value in any attention query, it produces no associated output; instead, $\mathbf{N}^{(0)}$ serves as the effective negative prompt tokens passed to the subsequent MLP layer and into the next attention layer, where it will be flipped again. Note that $N^{(0)}$ used as queries, keys, and values while $N^{(1)}$ is only used as keys and values.

This setup allows updates to the negative prompt based on attention from the image and from itself, while keeping the unflipped form active in the MLP path. It also prevents interference between positive and negative prompts and ensures that the flipped negative content affects only the intended image-to-negative attention path.

3.3. Attention Bias

We observe that even when the scaling factor $\alpha = 0$, including the negative prompt in the sequence still sometimes reduces image quality. This could be because the negative prompt "distracts" the image tokens' attention from the image tokens or positive prompts. To mitigate this effect, we introduce a negative bias $-\beta$ into the attention $\mathbf{I} \to \mathbf{N}^{(1)}$, thereby reducing the influence of the negative prompt.

doi.org/10.32388/POY29B

Figure 5. Effects of guidance scale (α) and attention bias (β) in image generation. Positive prompt is "a cat making a cake in the kitchen, the cat is wearing a chef's apron..." and negative prompt is "chef hat."

3.4. Padding Removel

In most models from Huggingface Diffusers[24], padding tokens in the text input are typically not masked during attention. This is likely because the models have learned to ignore padding, and masking them would add unnecessary overhead (due to some attention implementations like FlashAttention-2[25] that do not support arbitrary masking). However, when we invert the sign of the padding tokens, it degrades output quality. This could be because, although these tokens carry no semantic meaning, the sign-flipping introduces unseen states into the attention mechanism. To mitigate this, we remove padding tokens from the negative prompt embeddings. For the positive prompt, we retain padding tokens, as they do not introduce novel tokens and can improve generation quality. This aligns with training conditions and may allow the model to use padding positions as registers for auxiliary information.

4. Experiments

4.1. Dataset

Following Park et al.[8], we use ChatGPT o3[26] to generate pairs of prompts and negative prompts. Unlike prior work, our prompts are intentionally more challenging: the negative prompt is typically related to the positive one, and as a critical component—e.g., the positive prompt of a bike could have a negative prompt of "wheels". Additional examples are shown in Figure 7. Besides prompts, two questions are generated at the same time for later evaluation, one query if the image has the main object, either with or without the negative element, and the other one queries if the negative prompt element is missing. Prompts are generated in batches. Due to the fact that the model might output similar concepts for different prompts, it may introduce some repetition across batches or within batches with different phasing. There are 200 prompts generated, and we run them with 2 different seeds for the main results.

4.2. Baseline

We chose NAG[7] and NASA Nguyen et al.[15] as our baseline. We also used a base model without negative guidance as a bare baseline, aiming to show the lower bound of the dataset (i.e., how likely the positive prompt will introduce the negative concept, if there is no negative guidance). Because NASA's original source code was not publicly available at the time of writing, we reimplemented it based on NAG's codebase. Specifically, we replaced the guidance equation from NAG (Eq. 4) with NASA's equation (Eq. 3), removed normalization and blending, and enabled guidance when the scale is greater than 0 (instead of 1). Additionally, to compare our method in few-step models with CFG on original non-few-step models, we also used the original Stable Diffusion 3.5 Large with CFG as a baseline.

4.3. Metric

Following Park et al.[8]; Wei et al.[27], we used multimodal large language models (MLLM), specifically llama-4-maverick-17b-128e-instruct-fp8 (llama), to evaluate if the generated image follows the positive prompt and the negative prompt using the two questions generated during prompt generation. LLaMA 4 Maverick has a very high image reasoning MMMU mmmu score, higher than Gemma 3 and even GPT-4o gpt4o. We avoided using the same model (o3) for both evaluation and generation for cost control and to avoid bias within a model. We did not evaluate the quality of the generated images using models like ImageReward[28] or HPSv2 Wu et al.[29] as in NASA or NAG, as current quality or human preference

assessment models do not account for negative prompts. Removing a key element from the positive prompt (e.g., removing the roof from a house) is likely to reduce perceived quality, since the result deviates from what is considered "normal," even though that is the intended outcome. Both ImageReward and HPSv2 are built on top of image-text alignment models (CLIP[30] or BLIP[31]), which will likely lead to a decreased score when the main object is missing a critical part. Thus, we also let the MLLM rate the image quality from 0-1 for each image and told it to ignore the abnormality of following the negative prompt. Given that it is very hard to quantify the performances of methods in these cases, we used qualitative methods as the main evaluation and comparison, following Wang et al.[23].

4.4. Hyper-parameter Tunning

Although NAG[7] also targeted negative concept avoidance, its primary focus was on its effects on improving generation quality (using words like "blurry" or "low quality" as a negative prompt). We believe the hyperparameters reported in their work were tuned with an emphasis on quality rather than negation handling. Therefore, we re-tuned their hyperparameters moderately and manually targeting guidance scale (ϕ), blending factor (α), and normalization factor (τ). We will report experimental results on both original NAG (noted as NAG) and the improved hyperparameter version (noted as NAG++). The final hyperparameters used are $\phi = 11, \alpha = 0.5, \tau = 5$. This pushes the NAG to the edge of acceptable visual quality. Although we cannot ensure this is the best performing hyperparameters, we believe this version of NAG has better negative prompt following, with sacrifices in quality. These parameters are not swept to avoid over-fitting. However, we did conduct experiments on the positive-negative-quality trade-off study later in this paper.

4.5. Results

Quantitative results from MLLM as judge evaluation are shown in Table 1, and qualitative results are shown in Figure 7. P-value for the negative score is tested with the McNemar test. Qualitative results are from 10 randomly selected samples.

It is important to highlight that the MLLM assigns relatively generous quality scores; significantly distorted images may still receive high ratings. Empirically, we observed that images scoring around 60 are heavily distorted and exhibit numerous artifacts. For example, the image on the left of Figure 6 has a quality score of 70 yet displays severe degradation, and the image in the middle has noticeable distortion

but is rated with 90, and the image on the right has minor artifacts (meaning it is not perfect) while receiving a score of 100.

Figure 6. An example of a completely distorted image gets a relatively high quality score. The left one has a score of 70, the middle one has a score of 90, and the right one is a slightly distorted image, but still rated for 100.

doi.org/10.32388/POY29B

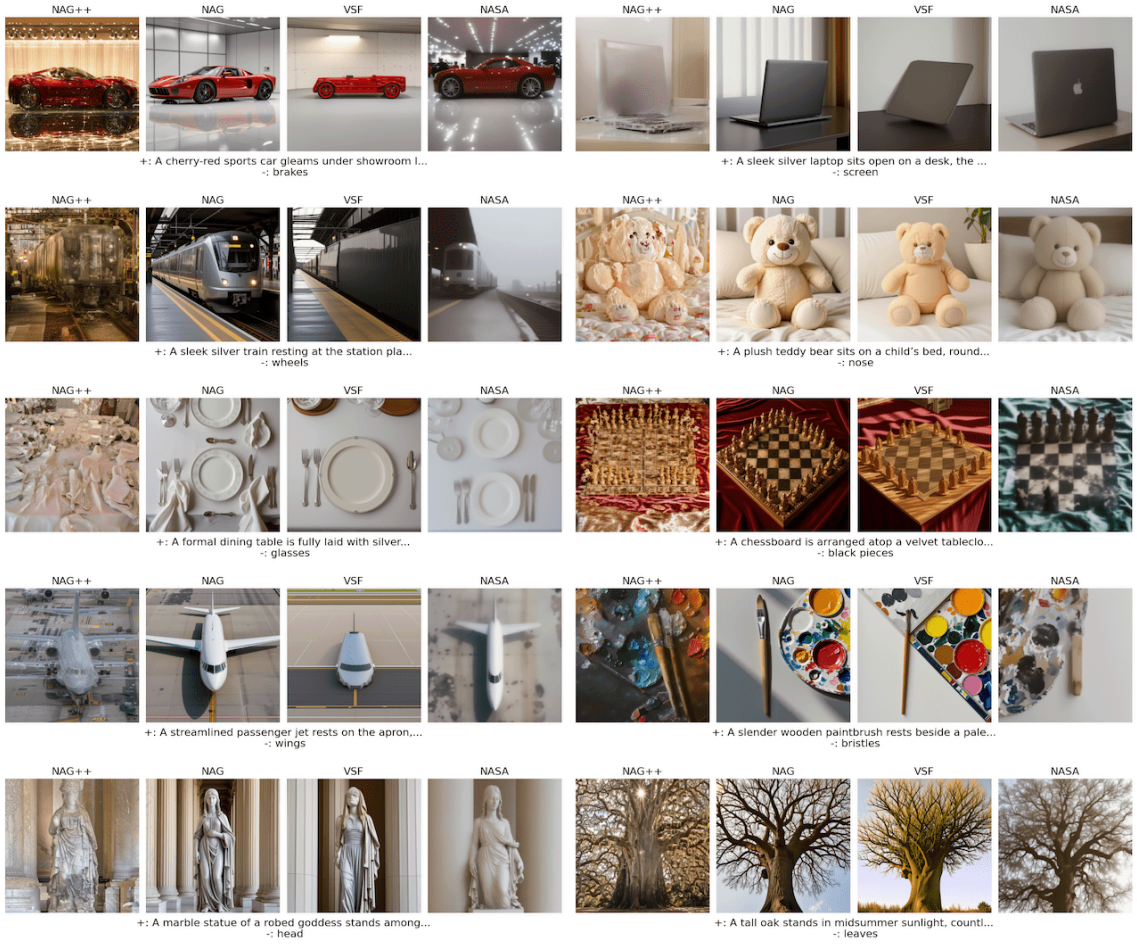


Figure 7. Qualitative comparison of an NAG version whose hyperparameter was tuned for negative guidance (NAG++), original NAG (NAG), and our method (VSF). These samples are randomly selected from the results.

ChatGPT said:Based on the quantitative results, VSF shows a significantly higher negative score than other methods, while maintaining comparable or better quality scores. However, its positive score is considerably lower. The trade-off between positive and negative scores is discussed in the extended section. Our method achieves a higher negative score than traditional CFG in non-few-step models, demonstrating a stronger ability to avoid negative elements even relative to the established strong baseline.

From the qualitative results, we can see that in many cases, our VSF avoided the negative prompt better, such as the plane wings and bristles examples. We can also see that in many cases, the guidance scale of NAG++ and our version of NASA is pushed to the limit of acceptable quality, yet still fails to follow the negative prompt. For VSF, the quality is between NAG and NAG++, sometimes resulting in very simple,

doi.org/10.32388/POY29B

lacking detail images like the red car, a jet without wings, or the laptop with keyboard. However, in many cases it still preserved high quality, such as the color palette, the chess, and the train. All three methods resulted in some unnatural physical formation, such as the weird location of the laptop hinge. There are also cases where all methods failed, such as removing the head from the statue and removing the nose from the teddy bear, indicating the significant challenges of our dataset. Additionally, our model sometimes removes more features than we want, such as the keyboard in the laptop and the windshield in the car, due to attention map dispersion or semantic similarity between these elements. Sometimes, VSF performs a bit worse than other methods, like in the leaves example.

4.6. Trade Off Curve

To systematically evaluate how effectively each model balanced positive prompt adherence, negative prompt adherence, and image quality, we conducted a hyperparameter sweep across each model. Specifically, we performed 66 runs for VSF and 287 runs for NAG, and 10 runs for NASA, with respect to their hyperparameter counts (2 for VSF, 3 for NAG, and 1 for NASA). A random sweep was executed besides for NASA, on which a grid search is used, and evaluations were conducted using Llama 4 (llama4) following the same criteria as previously described. Due to the large volume of runs, we limited our evaluation to the first 100 prompts with a single generation seed, potentially resulting in minor differences from earlier outcomes.

For the trade-off plot, runs were sorted by negative prompt adherence scores from highest to lowest. As we trace from the highest negative score to the lowest, we sequentially record the highest positive prompt or image quality scores encountered. Each run was marked as a critical point if it improved upon previously recorded positive prompt or quality scores. All critical points are plotted and connected, and shown in Figure 8.

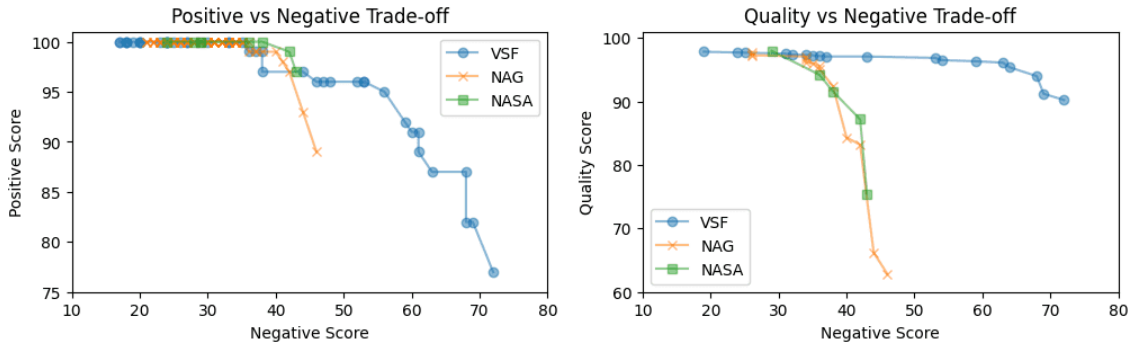


Figure 8. Trade off plot of positive-negative score and quality-negative score. Both axies follows "higher is better."

From both plots, we observe that as the negative score increases, NAG and NASA both exhibit a significantly steeper decline compared to VSF in both positive and quality scores. NASA has a steeper decline in quality compared to NAG, which is expected as NASA does not have the normalization and blending. In terms of positive score, VSF maintains scores above 90 even when the negative score rises to approximately 60. Regarding image quality, VSF similarly retains scores above 90 until a negative score of around 60, after which quality declines. In contrast, NAG and NASA both experience a sharp decline, with their quality score rapidly dropping to nearly 60 even before the negative score reaches 50.

Additionally, VSF demonstrates a broader operational range in negative scores. When necessary, it can achieve negative scores exceeding 70 while still preserving acceptable positive prompt adherence and image quality. Conversely, NAG and NASA become unacceptable in quality at negative scores below 50, limiting their practical effectiveness. Keep in mind that the MLLM usually overestimates the quality, and if an image is rated 60, it is usually completely distorted. See Figure 6 for example.

4.7. Attention Maps

Since our proposed method performs adaptive steering based on a negative attention map, we visualize the attention maps generated during the diffusion process in Figure 4. Extracting the full attention maps is difficult because efficient implementations, such as FlashAttention, do not explicitly store these maps, and storing and computing them will require a large amount of memory. Therefore, we computed only the unnormalized attention values between the image tokens and negative prompt tokens.

Figure 4 demonstrates that when the scale is set to 0, umbrellas appear, whereas setting the scale to 3 effectively removes them. As indicated in the attention maps, image tokens corresponding to regions where umbrellas might exist (e.g., above human heads) exhibit higher attention toward the negative prompt tokens. Specifically, in steps 4 and 5, regions above the individuals on the left and right show strong negative attention, aligning with areas visually identified as umbrellas. In the final image, these highlighted regions no longer contain umbrellas, confirming that our method effectively suppresses the presence of undesired objects at specific locations.

4.8. Ablation Study

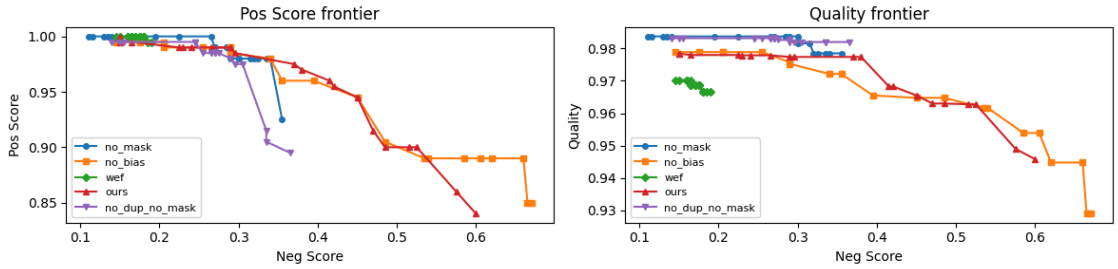


Figure 9. Trade Off Plot For Abliation Study

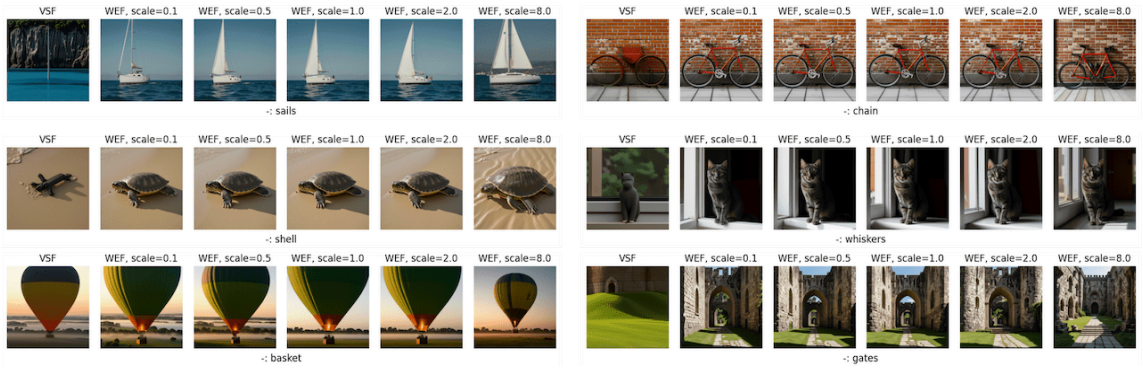


Figure 10. Example of Whole Embedding Flip (WEF), where the negative prompt embedding got flipped and concated with the positive prompt embedding before sending into a normal DiT

To evaluate the effectiveness of each component of our approach, we conducted an ablation study using the following settings. For each setting, we scanned across scales for all 200 prompts using the same seed. Similar to before, we plotted the trade-off curve for each setting.

Rather than altering the attention values, we explored a simpler and more intuitive approach: flipping the text embedding prior to input into the DiT (Whole Embedding Flip, WEF). This is similar to applying the CFG on text embeddings in NASA, but keeps the positive and negative tokens separated. Specifically, the negative text embedding is scaled by $-\alpha$, concatenated with the positive prompt embedding in the sequence length dimension, and used as the prompt embedding for the DiT. We did not remove the padding for the negative prompt, as we found out that removing it causes the negative prompts to have no effects at all. We also tested our approach with no bias, no mask (but still duplication), and no duplication, no mask. The trade-off plot is shown in Figure 9. We specifically show the results from WEF in Figure 10 across different scales. This simpler and more intuitive approach appears to have no effect. We hypothesize that this is because it is similar to flipping both the key and the value, causing regions most similar to the flipped key (i.e., least similar to the original negative prompt) to be pushed away, rather than pushing away regions most similar to the original negative prompt (i.e., unflipped key). From the figure, we can see that the configurations without masking have a sharp positive score drop as the negative score increases. The WEF has a very limited range of negative scores, confirming the qualitative results. Our methods and the one without attention bias has similar results, showing that the bias is optional and can be compensated with scaling changes.

	Positive Score	Negative Score	Quality Score
VSF	0.870	0.545	0.952
NAG[7]	0.993	0.220 ($p < 10^{-6}$)	0.968
NAG++	0.975	0.320 ($p < 10^{-6}$)	0.901
NASA[15]	0.970	0.380 ($p < 10^{-6}$)	0.867
None	0.990	0.195 ($p < 10^{-6}$)	0.968
CFG[12] **(non-few-step)**	1.000	0.300 ($p < 10^{-6}$)	0.956

Table 1. Positive scores (how well the model follows the positive prompts) and negative scores (how well the model avoids the negative prompts) of our model (VSF), NAG[7], and NAG with hyperparameter re-tuned (NAG++).

4.9. Adapting to Other DiT Models

In this paper, we primarily use SD-3.5 (8) due to simplicity and elegant architecture. However, our method can theoretically be adapted to any transformer-based diffusion or flow-matching model. To demonstrate this adaptability, we implemented our method on Wan 2.1 with CausVid LoRA[5][14].

For Wan 2.1, which uses cross-attention between image and text, masking is unnecessary and not used. Because our approach does not perform extrapolation and solely provides negative guidance, it cannot enhance overall quality significantly or replace CFG sampling in non-disstilled models, making it incompatible with the original Wan 2.1 model. Instead, we utilize CausVid[5], which enables Wan to function effectively without classifier-free guidance in few-step settings. Specifically, we used a LoRA[14] distilled from the original CausVid that can be directly applied on top of Wan 2.1. For qualitative results from Wan, please see the appendix.

We also tested our method on Flux Schnell[1]. However, due to its architecture combining one-stream and two-stream attention mechanisms, our approach did not significantly impact its tendency to ignore negative prompts. Future work should investigate these differences and explore ways to improve effectiveness.

4.10. Computational Cost

Since our method does not require two passes through the entire model (as in CFG) or the attention module (as in NAG or NASA), and only slightly increases the sequence length, its theoretical computatio nal cost is significantly lower, close to that of a single pass. However, due to implementation limitations (specifically, FlashAttention-2's lack of support for arbitrary attention masking), the actual runtime of our method is higher than the original single-pass MM-DiT models, and similar to NAG or NASA but much lower than CFG.

	Wan		SD3.5	
	Time	**VRAM**	**Time**	**VRAM**
Baseline	23.10s	22.05GB	2.14s	28.49GB
NASA	-	-	2.89s	28.50GB
NAG	25.58s	22.06GB	2.98s	28.50GB
VSF	22.70s	23.05GB	3.00s	28.53GB
VSF (No mask/bias)	22.70s	22.05GB	-	-

Table 2. The computation cost of each model. Time is measured in total runtime per sample, and VRAM is the peak RAM during the 25 samples generation. Since VSF Wan does not require a mask, and it is only used for bias, we also tested it without the bias. The SD3.5 model used is SD-3.5-Large-Turb,o and the Wan model used is Wan-2.1-T2V-1.3B.

To accurately measure the computational cost, we evaluate the runtime of 25 identical prompts under four settings: no guidance, NAG, NASA, and our proposed guidance, VSF, and then report the average runtime and peak memory usage for each setting. To avoid GPU thermal throttling affecting the results, we pause for at least 5 minutes between each set of tests. The tests are done on NVIDIA A100 40GB on Google Colab, as this is the most accessible option for high-end GPUs for users. Stable Diffusion Turbo is generated in 8 steps for 1024x1024 resolution, Wan is generated in 8 steps with 480x832 resolution, and 81 frames. The results are shown in Table 2.

From the table, VSF requires marginally more time and memory than NAG in SD3.5, while they are both significantly faster than theoretical CFG time, which would be twice the baseline. In Wan, VSF outperforms NAG and is even slightly better than the baseline (likely due to nature variation) in terms of compute time, though it consumes 1GB more memory, likely due to the attention bias being stored. Since this bias is optional, we tested VSF Wan's performance with it removed, which results in an improvement in VRAM usage such that it uses the same amount of VRAM as baseline and NAG, and no change in runtime.

5. Conclusion and Future Work

In this paper, we introduced VSF, a novel approach for enhancing negative prompt adherence in image and video generation models. Our method involves flipping the sign of attention values corresponding to negative prompts, effectively suppressing unwanted content. Experimental results indicate that VSF significantly outperforms previous methods, NAG Chen et al.[7], in terms of negative prompt adherence, with only minor trade-offs in overall quality and positive prompt fidelity. VSF also only has one main hyperparameter and one minor hyperparameter, making it easier to tune them in downstream tasks.

Future work may involve extending VSF to other architectures, such as Flex Schnell, improving robustness through normalization and blending techniques similar to those employed by NAG, and optimizing computational efficiency by using a better attention implementation. Additionally, conducting a larger-scale human evaluation study would help mitigate inaccuracies observed in MLLM-based assessments. Investigating the attention maps and diffusion trajectories of our model could further elucidate the underlying mechanisms of VSF.

Appendix

A.1. Qualitative Results for Wan

To qualitatively evaluate the effects of our method, we present several example generations across different models. These samples are intended for visualization, and results may vary with different hyperparameters or sampling conditions. The examples demonstrate both strengths and limitations across models and are not meant to reflect comprehensive performance.

In the first example, we evaluate cutting tomatoes on a board, where the negative prompt is a wooden board. The original video (generated without negative guidance) includes a wooden board. However, the NAG and VSF outputs both replace it with non-wooden surfaces—a plastic or glass-like surface. However, both original and NAG outputs exhibit unnatural physics during the cutting motion. The VSF sample avoids major unnatural physics but fails to preserve the shape of the tomato, as the tomato deforms unrealistically during slicing. These issues likely arise from limitations in the small 1.3B parameter model since it also appears in the original video.

The second example involves a lava river without glowing. In still frames, the outputs from both NAG and VSF appear similar to regular rivers, but in motion, they exhibit texture and flow characteristics

resembling lava. The NAG sample, however, leans more toward a natural river, reflecting weaker adherence to the intended appearance.

doi.org/10.32388/POY29B

+: A knief cutting a tomato on a cutting board, the camera captures the knife's sharp edge slicing thro...
-: wooden board, low quality, blurry, low resolution, weird motion

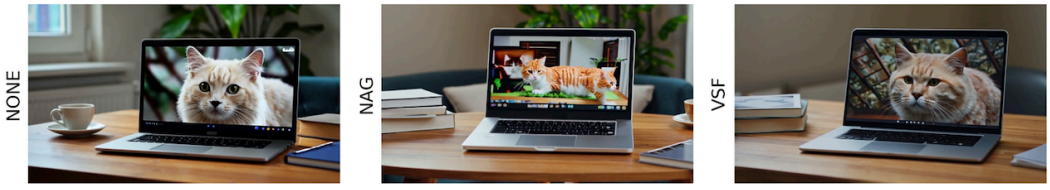
+: A lava river flowing through a volcanic landscape, dark rocky terrain. The camera captures the the f...
-: red hot, bright, glow, low quality, blurry, low resolution, weird motion

+: A plane flying over a snowy mountain range, with the sun setting in the background. The camera captu...
-: wings, low quality, blurry, low resolution, weird motion

+: A machine learning scientist working in a lab, analyzing data on a computer screen. The camera captu...
-: male with glasses, low quality, blurry, low resolution, weird motion

+: A pet running through a field of flowers, with the sun shining brightly. The camera captures the pet...
-: dog, low quality, blurry, low resolution, weird motion

+: A cat chef is cooking a delicious meal in a cozy kitchen, with the camera capturing the cat's focuse...
-: window, low quality, blurry, low resolution, weird motion



+: A laptop is on the table, playing a video of a cat. The laptop is silver and sleek, with a high-reso...
-: keyboard, low quality, blurry, low resolution, weird motion

doi.org/10.32388/POY29B

Figure 11. Example of Wan 2.1 with no negative guidance (NONE), NAG guidance (NAG), and our guidance (VSF). Positive and negative prompts are shown at the bottom of each figure.

In the third example, the prompt requests a plane with no wings. The original outputs clearly retain wings. The NAG output reduces the wing size but does not remove it completely. VSF most closely satisfies the constraint, with wings nearly absent, indicating a stronger capability in element removal.

The fourth example aims to remove bias in generating a machine learning scientist by excluding the common depiction of a male with glasses. Both NAG and VSF successfully eliminate the targeted attributes. Among them, NAG produces the more natural-looking output, with better visual coherence.

In the fifth case, the prompt asks for a pet running through flowers, explicitly excluding dogs. The original sample resembles a dog-like hybrid. The NAG version trends more cat-like and adheres better to the constraint. The VSF sample avoids the dog but results in a character resembling a fictional figure from a children's story, making its alignment with the "pet" concept weaker.

In the sixth example, we request a cat chef cooking in a kitchen without a window. Since the original sample already lacks a window, it does not test model behavior under the constraint. In the NAG output, under such a high guidance scale, it introduces undesired concepts, such as a human inside the cat adding seasoning to the pan, suggesting instability under strong steering.

The final example prompts for a laptop playing a cat video without a keyboard. Both NAG and VSF fail to remove the keyboard, likely due to its strong association with the laptop concept. Additionally, the NAG output results a cat with two heads in the screen, which may stem from the high guidance scale or the model's limited capacity.

From these examples, we observe that both models succeed and fail in different scenarios. VSF is more effective at removing explicit elements (e.g., wings) but less reliable at excluding abstract sub-concepts (e.g., dog from pet, male from scientist). Overall, the outputs of VSF and NAG are comparable. However, our method (VSF) operates at a lower compute cost, as discussed in the following section.

Statements and Declarations

Acknowledgements

This work was supported by the NFRF under grant GR024801 and the CFI under grant GR024473. We also acknowledge Weathon Software (https://weasoft.com) for providing computing credits via Google Colab, and Lambda, Inc. (https://lambda.ai/) for computing credits via Lambda Cloud and Lambda Inference.

References

1. ^a, b, c, d, e*Black Forest Lab (2025). "Black-forest-labs/FLUX.1-schnell · Hugging Face." Hugging Face. https://huggingface.co/black-forest-labs/FLUX.1-schnell.*
2. ^*Woolf M (2022). "Stable Diffusion 2.0 and the Importance of Negative Prompts for Good Results." minimaxir. https://minimaxir.com/2022/11/stable-diffusion-negative-prompt/.*
3. ^a, b, c, d, e*Stability AI. "Introducing Stable Diffusion 3.5." Stability AI. https://stability.ai/news/introducing-stable-diffusion-3-5.*
4. ^*Wan Team, Wang A, Ai B, Wen B, Mao C, Xie C, Chen D, Yu F, Zhao H, Yang J, Zeng J, Wang J, Zhang J, Zhou J, Wang J, Chen J, Zhu K, Zhao K, Yan K, Huang L, Feng M, Zhang N, Li P, Wu P, Chu R, Feng R, Zhang S, Sun S, Fang T, Wang T, Gui T, Weng T, Shen T, Lin W, Wang W, Wang W, Zhou W, Wang W, Shen W, Yu W, Shi X, Huang X, Xu X, Kou Y, Lv Y, Li Y, Liu Y, Wang Y, Zhang Y, Huang Y, Li Y, Wu Y, Liu Y, Pan Y, Zheng Y, Hong Y, Shi Y, Feng Y, Jiang Z, Han Z, Wu Z, Liu Z (2025). "Wan: Open and Advanced Large-Scale Video Generative Models." arXiv. doi:10.48550/arXiv.2503.20314.*
5. ^a, b, c, d, e*Yin T, Zhang Q, Zhang R, Freeman WT, Durand F, Shechtman E, Huang X (2025). "From Slow Bidirectional to Fast Autoregressive Video Diffusion Models." arXiv. doi:10.48550/arXiv.2412.07772.*
6. ^a, b, c, d, e, f*Schramowski P, Brack M, Deiseroth B, Kersting K (2023). "Safe Latent Diffusion: Mitigating Inappropriate Degeneration in Diffusion Models." arXiv. doi:10.48550/arXiv.2211.05105.*
7. ^a, b, c, d, e, f, g, h, i, j*Chen D, Bandyopadhyay H, Zou K, Song Y (2025). "Normalized Attention Guidance: Universal Negative Guidance for Diffusion Models." arXiv. doi:10.48550/arXiv.2505.21179.*
8. ^a, b, c, d, e, f, g*Park J, Lee J, Song J, Yu S, Jung D, Yoon S (2025). "Know "No" Better: A Data-Driven Approach for Enhancing Negation Awareness in CLIP." arXiv. doi:10.48550/arXiv.2501.10913.*
9. ^a, b, c*Alhamoud K, Alshammari S, Tian Y, Li G, Torr P, Kim Y, Ghassemi M (2025). "Vision-Language Models Do Not Understand Negation." arXiv. doi:10.48550/arXiv.2501.09425.*

10. [a], [b], [c]*Singh J, Shrivastava I, Vatsa M, Singh R, Bharati A (2025). "Learning the Power of "No": Foundation Models with Negations." In 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 8002–8012. doi:10.1109/WACV61041.2025.00777.*

11. [a], [b], [c]*Singh J, Shrivastava I, Vatsa M, Singh R, Bharati A (2024). "Learn "No" to Say "Yes" Better: Improving Vision-Language Models via Negations." arXiv. doi:10.48550/arXiv.2403.20312.*

12. [a], [b], [c]*Ho J, Salimans T (2022). "Classifier-Free Diffusion Guidance." arXiv. doi:10.48550/arXiv.2207.12598.*

13. [a], [b]*Lin S, Wang A, Yang X (2024). "SDXL-Lightning: Progressive Adversarial Diffusion Distillation." arXiv. doi:10.48550/arXiv.2402.13929.*

14. [a], [b], [c], [d]*Seppanen J. Kijai/WanVideo comfy · Hugging Face. https://huggingface.co/Kijai/WanVideo_comfy.*

15. [a], [b], [c], [d], [e], [f], [g]*Nguyen V, Nguyen A, Dao T, Nguyen K, Pham C, Tran T, Tran A (2024). "SNOOPI: Supercharged One-step Diffusion Distillation with Proper Guidance." arXiv. doi:10.48550/arXiv.2412.02687.*

16. [a], [b], [c], [d], [e], [f]*Koulischer F, Deleu J, Raya G, Demeester T, Ambrogioni L (2025). "Dynamic Negative Guidance of Diffusion Models." arXiv. doi:10.48550/arXiv.2410.14398.*

17. [a], [b], [c], [d]*Ban Y, Wang R, Zhou T, Cheng M, Gong B, Hsieh C (2024). "Understanding the Impact of Negative Prompts: When and How Do They Take Effect?." arXiv. doi:10.48550/arXiv.2406.02965.*

18. [a], [b]*Yuksekgonul M, Bianchi F, Kalluri P, Jurafsky D, Zou J (2023). "When and why vision-language models behave like bags-of-words, and what to do about it?." arXiv. doi:10.48550/arXiv.2210.01936.*

19. [^]*Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ (2023). "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer." arXiv. doi:10.48550/arXiv.1910.10683.*

20. [^]*Esser P, Kulal S, Blattmann A, Entezari R, Müller J, Saini H, Levi Y, Lorenz D, Sauer A, Boesel F, Podell D, Dockhorn T, English Z, Lacey K, Goodwin A, Marek Y, Rombach R (2024). "Scaling Rectified Flow Transformers for High-Resolution Image Synthesis." arXiv. doi:10.48550/arXiv.2403.03206.*

21. [^]*Lipman Y, Chen RTQ, Ben-Hamu H, Nickel M, Le M (2023). "Flow Matching for Generative Modeling." arXiv. doi:10.48550/arXiv.2210.02747.*

22. [^]*Koulischer F, Handke F, Deleu J, Demeester T, Ambrogioni L (2025). "Feedback Guidance of Diffusion Models." arXiv. doi:10.48550/arXiv.2506.06085.*

23. [a], [b]*Wang L, Li Y, Chen Z, Wang J, Zhang Z, Zhang H, Lin Z, Chen Y (2025). "TransPixeler: Advancing Text-to-Video Generation with Transparency." arXiv. doi:10.48550/arXiv.2501.03006.*

24. [^]*Platen P von, Patil S, Lozhkov A, Cuenca P, Lambert N, Rasul K, Davaadorj M, Nair D, Paul S, Liu S, Berman W, Xu Y, Wolf T. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers.*

25. ^Dao T (2023). "FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning." arXiv. doi:10.48550/arXiv.2307.08691.

26. ^Open AI. Introducing OpenAI o3 and o4-mini. https://openai.com/index/introducing-o3-and-o4-mini/.

27. ^Wei X, Zhang J, Wang Z, Wei H, Guo Z, Zhang L (2025). "TIIF-Bench: How Does Your T2I Model Follow Your Instructions?." arXiv. doi:10.48550/arXiv.2506.02161.

28. ^Xu J, Liu X, Wu Y, Tong Y, Li Q, Ding M, Tang J, Dong Y (2023). "ImageReward: Learning and Evaluating Human Preferences for Text-to-Image Generation." arXiv. doi:10.48550/arXiv.2304.05977.

29. ^Wu X, Hao Y, Sun K, Chen Y, Zhu F, Zhao R, Li H (2023). "Human Preference Score v2: A Solid Benchmark for Evaluating Human Preferences of Text-to-Image Synthesis." arXiv. doi:10.48550/arXiv.2306.09341.

30. ^Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, Sastry G, Askell A, Mishkin P, Clark J, Krueger G, Sutskever I (2021). "Learning Transferable Visual Models From Natural Language Supervision." arXiv. doi:10.48550/arXiv.2103.00020.

31. ^Li J, Li D, Xiong C, Hoi S (2022). "BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation." arXiv. doi:10.48550/arXiv.2201.12086.

Declarations

Funding: This work was supported by the NFRF under grant GR024801 and the CFI under grant GR024473. We also acknowledge Weathon Software (\url{https://weasoft.com}) for providing computing credits via Google Colab, and Lambda, Inc. (\url{https://lambda.ai/}) for computing credits via Lambda Cloud and Lambda Inference.

Potential competing interests: No potential competing interests to declare.