Tiny Inference-Time Scaling with Latent Verifiers

Davide Bucciarelli[*1,2] Evelyn Turri[*1] Lorenzo Baraldi[2],
Marcella Cornia[1] Lorenzo Baraldi[1] Rita Cucchiara[1]
[1]University of Modena and Reggio Emilia, Italy [2]University of Pisa, Italy
[1]{name.surname}@unimore.it, [2]{name.surname}@phd.unipi.it
aimagelab.github.io/VHS

Abstract

*Inference-time scaling has emerged as an effective way to improve generative models at test time by using a verifier to score and select candidate outputs. A common choice is to employ Multimodal Large Language Models (MLLMs) as verifiers, which can improve performance but introduce substantial inference-time cost. Indeed, diffusion pipelines operate in an autoencoder latent space to reduce computation, yet MLLM verifiers still require decoding candidates to pixel space and re-encoding them into the visual embedding space, leading to redundant and costly operations. In this work, we propose **V**erifier on **H**idden **S**tates (**VHS**), a verifier that operates directly on intermediate hidden representations of Diffusion Transformer (DiT) single-step generators. VHS analyzes generator features without decoding to pixel space, thereby reducing the per-candidate verification cost while improving or matching the performance of MLLM-based competitors. We show that, under tiny inference budgets with only a small number of candidates per prompt, VHS enables more efficient inference-time scaling reducing joint generation-and-verification time by 63.3%, compute FLOPs by 51% and VRAM usage by 14.5% with respect to a standard MLLM verifier, achieving a +2.7% improvement on GenEval at the same inference-time budget.*

1. Introduction

Diffusion and flow-based models [16, 25, 39] have recently transformed image synthesis, producing samples that closely resemble natural imagery with remarkable fidelity and controllability. However, their generation process remains computationally expensive and often misaligned with user intent. To mitigate these limitations, recent works have adopted the inference-time scaling paradigm [19, 30, 37, 47], which allocates additional computational budget at inference by generating multiple candidate samples and selecting the most suited among them. This framework relies on two key components: (i) an exploration algorithm that generates multiple candidates, and (ii) a verifier that assigns scores to the candidates and selects those that best match the prompt.

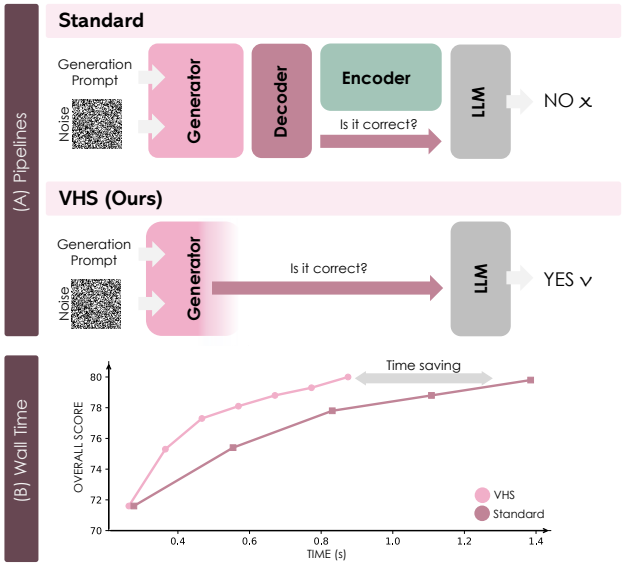

Figure 1. (A) Comparison between standard inference-time scaling and VHS. VHS skips part of the generation pipeline and avoids the decoding and re-encoding steps. (B) VHS achieves a comparable quality score on GenEval [10] in just 57% of the compute time.

In image generation, verifiers are typically implemented by fine-tuning Multimodal Large Language Models (MLLMs) [4] with an image scoring objective. Nevertheless, MLLMs are computationally heavy, and their inference cost is not negligible. Despite this, recent literature [19, 30] mostly accounts for the number of function evaluations (*e.g.*, diffusion steps), while treating the cost of the verifier as implicit overhead, leading to an incomplete view of the computational footprint of inference-time scaling. Moreover, many existing evaluations assume very large budgets, sometimes involving thousands of function evaluations [19], whereas practical deployment scenarios, such as commercial image generation services, typically operate under much tighter constraints, often returning only a handful of images (*e.g.*,

up to four candidates) per prompt. Further, visual generative models typically operate in a compressed latent space [34], defined by an autoencoder, whereas MLLMs rely on an external visual encoder (*e.g.*, CLIP [33]) to obtain image representations. Thus, to score a generated sample, the latent representation must be decoded into pixel space and then re-encoded by the visual backbone of the MLLM. Although this decode-encode overhead may be acceptable in standard multi-step generators, it becomes increasingly significant in the case of single-step image generators, which can produce images in a single function evaluation [7, 29, 36, 50].

Following these considerations, we argue that the architecture of MLLM-based verifiers [13, 47] should be reconsidered in light of the specific characteristics of the task. To this aim, we introduce **V**erifier on **H**idden **S**tates (**VHS**), a MLLM-based verifier that directly aligns internal hidden representations of image generators with the embedding space of an LLM. Concretely, VHS operates on single-step image generators, extracting latent features during the generation process, and uses these hidden states as the visual inputs to the LLM (Fig. 1). This way, VHS eliminates the encoding-decoding overhead in the evaluation step, enabling significantly more efficient verification within the inference-time scaling framework, while retaining the expressivity of MLLM-based scoring. As a consequence, VHS is well-suited for tiny computational budgets, where only a small number of candidates per prompt is affordable, and thus closely aligns with the practical constraints and deployment settings of real-world commercial image generation services.

We evaluate VHS in terms of latency and verification quality on the GenEval benchmark [10]. In combination with the single-step generator SANA-Sprint [7] and a compact LLM (Qwen2.5-0.5B [1]), VHS reduces the joint generation-and-verification time by 63.3% of that required by a standard MLLM-based verifier. Furthermore, under matched wall-clock budgets, VHS improves inference-time scaling performance on GenEval, achieving overall score gains of 3.1%, 1.7%, and 0.5% over a CLIP-based MLLM verifier in the Best-of-2, Best-of-4, and Best-of-6 settings, respectively.

In summary, our main contributions are:

- We introduce VHS, a verifier that operates directly on internal hidden states of DiT-based image generators, aligning visual latents with an LLM without passing through pixel space or an external visual encoder.
- We define a latency-aware inference-time scaling setting for single-step image generation, explicitly measuring wall-clock time and analyzing performance in realistic few-sample generation regimes.
- We provide a thorough empirical study of verifier design and latency, comparing alternative architectures and configurations (*i.e.*, in terms of layers, backbones, and loss functions) and quantifying the trade-offs between computational cost and semantic alignment.

2. Related Work

Image Generation Techniques. Image generation has advanced substantially with the advent of diffusion models [16, 39], which have surpassed GANs [11] in both sample quality and training stability. Latent diffusion models [34] further extended this progress by operating in a compressed latent space, enabling high-resolution synthesis at a manageable computational cost. While early diffusion architectures relied primarily on U-Nets [35] for noise prediction, these have recently been surpassed by Diffusion Transformers (DiTs) [31], which offer improved scalability and performance. In parallel, flow-based approaches [9, 25] have reformulated the diffusion objective from noise estimation to velocity field prediction, providing an alternative yet closely related view of the generative process.

A complementary line of research focuses on improving inference efficiency. Few-step and even single-step diffusion models have been developed via distillation, making it possible to generate high-fidelity images with only a handful (or even a single) denoising step [7, 29, 36, 50]. In this area, Stable Diffusion XL-Turbo [36] introduced adversarial diffusion distillation to ensure high-fidelity synthesis in the low-step regime and leveraged large pre-trained multi-step models as teachers, with a mixture of adversarial training and score distillation. Subsequently, PixArt-α-DMD [50] proposed a distribution-matching distillation approach to align the student with the teacher model at the distribution level. Differently, SANA-Sprint [7] presented a hybrid distillation framework that combines training-free continuous-time consistency distillation with latent adversarial distillation, and enables efficient adaptation of pre-trained diffusion or flow-matching models in the few-step generation scenario.

Inference-Time Scaling. Inference-time scaling [38] consists in allocating additional computational resources during inference to improve model performance, rather than increasing compute during training. This strategy, widely adopted in NLP for LLM inference [12, 38, 46], has been recently extended to visual content generation [2, 19, 30, 37, 47]. In this context, it has been shown that allocating more compute time, beyond simply increasing the number of diffusion steps [30], can significantly enhance generation quality.

Inference-time scaling methods for visual generation typically rely on two main components: a search algorithm and a verifier. The former generates candidate samples, while the latter evaluates and ranks them to select the best output. The simplest strategy, Best-of-N, independently samples and scores N candidates, selecting the highest-scoring one as the final result. Another algorithm, widely adopted in LLM inference-time scaling is beam search [38], a heuristic algorithm that maintains the top-k most probable candidates at each step, balancing exploration and efficiency to improve generation quality over greedy sampling.

On the other end, verifiers are often based on MLLMs, leveraging their ability to interpret complex prompts and assess visual-textual alignment in the generated content. For instance, VQA-Score [24] employs a Visual Question Answering model that scores samples based on the probability of the "yes" token in response to predefined questions assessing prompt fulfillment. Similarly, Vision-Reward [49] queries an MLLM with fine-grained binary questions and combines the results through a learned weighting scheme.

In contrast with previous literature, we propose a verifier that directly works in the latent space of the generator, significantly reducing the computational overhead of verification.

Multimodal Large Language Models. MLLMs extend traditional language models by integrating information across multiple modalities [4, 22, 26, 27, 44], most notably, vision and text. Common architectures rely on a pretrained image encoder [8, 33, 51] whose embeddings are projected into the input space of the LLM through a lightweight adapter. This design allows the visual features to be integrated seamlessly into the token sequence of the LLM, enabling multimodal understanding and grounded generation.

This framework was popularized by LLaVA [26, 27], which employed simple linear layers as connector and introduced a two-stage training pipeline: aligning the connector using image-caption pairs, and subsequently fine-tuning the entire model on instruction-following datasets. Building on this, several works have proposed to improve visual grounding and fine-grained alignment. Idefics3 [20] partitions images into spatial tiles encoded independently, improving localization and detailed perception. Similarly, Qwen2.5-VL [1] incorporates 2D positional encodings into token representations to better preserve spatial structure within images. In contrast, we directly align hidden states of a DiT-based generator with the LLM, enabling image evaluation from latent representations rather than decoded pixels.

3. Proposed Method

3.1. Preliminaries

The objective of our approach is to assess content quality directly from the latent representations of a single-step image generator. In the following, we first formalize the generative process of multi-step models and subsequently introduce the single-step formulation adopted in our method.

Visual generative models, such as diffusion [16, 34] and flow-based models [9, 25, 47], synthesize data through a multi-step refinement process. Starting from a latent variable sampled from a prior distribution, $z_T \sim p_T$ (*e.g.* $\mathcal{N}(0, \mathbf{I})$), the model progressively refines it into a structured representation z_0 by constructing a discrete trajectory

$$z_T \to z_{T-1} \to \cdots \to z_1 \to z_0, \tag{1}$$

where transition steps are parameterized by a neural network f_θ that predicts model-specific quantities such as noise or velocity, depending on the underlying framework. This iterative process transports samples from the prior p_T toward an approximation of the target distribution p_{data}, yielding a sequence $\{z_t\}_{t=T}^{0}$ that we refer to as the *generative trajectory*.

Concretely, in DiT-based [9, 31] generators, the noisy latent z_t is processed by a Transformer backbone that produces a sequence of hidden representations $\{h_\ell\}_{\ell=0}^{L-1}$ generated with $h_\ell = \mathrm{DiT}_\ell(h_{\ell-1}, t)$, where L is the number of layers in the DiT and h_0 is the noisy latent z_t. Lastly, the DiT layers are followed by a decoder $\mathcal{D}$ that operates on the final hidden state h_{L-1} after projection and normalization (*i.e.*, z_0). In both diffusion [34] and flow models, the generation trajectory is defined not in pixel space but in the compressed latent space of an autoencoder [6], which we define as $\mathcal{E}$. During sampling, the generative trajectory $\{z_t\}_{t=0}^{T}$ evolves entirely in $\mathcal{E}$, and the final image is obtained by decoding the terminal latent z_0 via $x_0 = \mathcal{D}(z_0)$.

In contrast, a single-step generator is obtained by distilling a standard diffusion or flow-based model into a network that maps a latent sample $z_T \sim \mathcal{N}(0, I)$ to an image in one forward pass producing a generative trajectory with $T = 1$. While in multi-step diffusion and flow-based models the computational cost of the decoding operator $\mathcal{D}$ is typically negligible compared to the iterative sampling process, in single-step generators [7, 36, 50] this balance shifts: the forward pass of $\mathcal{D}$ becomes a non-trivial component of the total inference cost. For this reason, our method operates directly on the intermediate latent representation h when tasked with verifying generated samples, thereby skipping the forward pass through $\mathcal{D}$ and avoiding any decoding overhead.

3.2. Latent Verifier

Within the inference-time scaling framework, a key component is the verifier model, which evaluates generated samples and identifies the most promising ones. In our formulation, we define the verifier as a model $\mathcal{S}_\theta$ that, given a generated sample x_0 and the user prompt p, outputs $s \in \{\text{Yes}, \text{No}\}$ indicating whether the sample is semantically aligned with p. Recent works [19, 21, 47, 52] typically implement verifiers using MLLMs. Although such models have shown strong performance in assessing generation quality, the computational cost associated with their scoring procedure is non-negligible. Nevertheless, most inference-time scaling studies [19, 30] quantify the computational budget solely by the number of generator function evaluations (*e.g.*, diffusion steps), with the cost of running the MLLM-based verifier either implicitly ignored or treated as negligible.

Formally, an MLLM-based verifier can be decomposed into three components: (i) a visual encoder $\mathcal{V}$, which maps an input image x_0 to a sequence of visual tokens; (ii) a connector $\mathcal{C}$, which projects these visual tokens into the embedding space of the language model; and (iii) a language model,

Figure 2. Comparison between a standard generation-verification pipeline (top) and VHS (bottom). VHS consumes visual features directly from the hidden states of the generator, bypassing subsequent DiT layers, autoencoder (AE) decoding, and CLIP-based re-encoding, significantly reducing sampling and verification overhead.

which performs multimodal reasoning over the concatenated visual and textual tokens and produces the final score. In the inference-time scaling setting, this architecture is used as follows: a latent sample z_0 is drawn from the latent space of the generator, decoded into pixel space as $x_0 = \mathcal{D}(z_0)$, and then processed by the verifier to produce a score

$$s = \mathcal{S}_\theta(z_0, p) = \mathrm{LLM}\big(\mathcal{C}(\mathcal{V}(\mathcal{D}(z_0))), p\big). \tag{2}$$

In this pipeline, $\mathcal{V}$ is responsible only for re-encoding visual information that has already been implicitly represented in the latent space of the generator [32, 41, 48]. We claim that for generative models that operate in a rich latent space, this additional pass through $\mathcal{V}$ is not semantically essential for the verification task. Instead, it does introduce a non-trivial decode-encode overhead: the latent z_0 must undergo two successive transformations, $\mathcal{D}(z_0)$ and $\mathcal{V}(\cdot)$, before the LLM can reason about the sample.

Our **V**erifier on **H**idden **S**tates (**VHS**) explicitly bypasses the decoding-encoding bottleneck by removing the visual encoder from the verification loop. Instead of operating on the decoded image, VHS directly consumes hidden representations from the generator. Specifically, VHS acts on the output of a DiT layer $\ell^\star \in \{0, \dots, L-1\}$, denoted as $h_{\ell^\star}$, and feeds it to the connector $\mathcal{C}$ of the MLLM, as follows:

$$s = \mathcal{S}_\theta(z_0, p) = \mathrm{LLM}\big(\mathcal{C}(h_{\ell^\star}), p\big), \tag{3}$$

where the connector $\mathcal{C}$ is trained to align $h_{\ell^\star}$ with the LLM input space, treating hidden features like image features.

This design yields two key advantages. First, it completely removes the decoding-encoding pipeline $z_0 \to x_0 \to \mathcal{V}(x_0)$ from the verification process, thereby reducing per-sample evaluation latency. Second, since VHS accesses hidden states at layer $\ell^\star$, it allows us to truncate the generator during verification and skip the remaining $L - (\ell^\star + 1)$ layers. As a result, VHS provides semantically informed verification at a fraction of the computational cost of standard MLLM-based verifiers, making inference-time scaling substantially more practical in low-latency generation regimes. An overview of our approach is shown in Fig. 2, in comparison with a standard generation-verification pipeline.

3.3. Training Procedure Overview

VHS is trained via a two-stage procedure. First, in an alignment stage, we adapt the visual representation from the generator hidden layers to be compatible with the LLM backbone. Second, we fine-tune the model as a verifier.

Alignment Stage. In this stage, the goal is to align the visual representations extracted from $h_{\ell^\star}$ with the representation space of the LLM. Unlike standard visual encoders, our visual embedder is a generative model. As a consequence, we first need to generate raw images to obtain the intermediate features $h_{\ell^\star}$ used for alignment. Concretely, we build upon the dataset used in the first stage of the LLaVA training [26], which provides image-caption pairs usually employed to train MLLMs. Starting from each caption, we employ the generator to produce a synthetic image and record the associated hidden representation $h_{\ell^\star}$. Notably, this may introduce inconsistencies between the original caption and the generated image, due to hallucinations or semantic drift in the generator. To mitigate this, we re-caption each generated image using Gemma-3-4B [42], and use the resulting captions as the textual supervision for the alignment stage.

Verifier Fine-tuning. While the alignment stage is largely consistent with standard MLLM training, the verifier fine-tuning stage explicitly adapts the model to the scoring objective required in the inference-time scaling setting. Building on existing literature [21], we adopt the prompts of the training dataset of Reflect-DiT [21] and generate 20 candidate

Table 1. End-to-end inference time, FLOPs and VRAM usage for Best-of-N generation with SANA-Sprint [7] under different computational budgets, along with the relative savings (%) compared to the standard verifier.

	Inference Time (ms)					TFLOPs					Peak VRAM Usage (GB)				
	Saved (%)	Bo1	Bo2	Bo4	Bo6	**Saved (%)**	Bo1	Bo2	Bo4	Bo6	**Saved (%)**	Bo1	Bo2	Bo4	Bo6
MLLM w/ CLIP	-	277	554	1108	1662	-	15.1	28.5	55.1	81.8	-	13.8	15.5	18.8	22.2
MLLM w/ AE	50.2%	138	401	677	953	51.0%	7.4	14.8	29.5	44.3	**14.5%**	**11.8**	**11.9**	**12.3**	**12.6**
VHS on h_7	**63.3%**	**102**	**363**	**565**	**767**	**62.9%**	**5.6**	**11.3**	**22.5**	**33.8**	**14.5%**	**11.8**	**11.9**	**12.3**	**12.6**

images per prompt, resulting in a total of 118k samples. These candidates are categorized by Gemma-3-4B [42] into the respective GenEval categories [10] and scored with its automatic evaluator. Based on these evaluation scores, we derive binary labels (Yes/No) for each image in the dataset.

Analysis of the training set and GenEval benchmark reveals a significant class imbalance, with correct samples substantially overrepresented. A uniform weighting scheme in the training loss consequently underemphasizes the minority "incorrect" class, leading to suboptimal verifier performance. To address this, VHS employs a weighted cross-entropy loss during verifier fine-tuning. This approach re-weights the training signal proportionally to class frequencies, effectively compensating for the skewed label distribution and improving model calibration on underrepresented samples.

4. Experimental Results

4.1. Implementation Details

In the alignment stage, we follow the LLaVA [26] training scheme, and tune only the newly initialized connector module during the first stage. Differently, in the verifier fine-tuning stage, we train both the connector and the whole language model, splitting the generated datasets in training (80%) and evaluation (20%), selecting the model yielding the best evaluation loss. All models follow the exact same training procedure, ensuring a fair comparison between models trained with equivalent data and policy.

To derive a more granular scoring mechanism from binary labels, we leverage the LLM output probability of the sampled token ("yes" or "no") to produce a continuous score. Best-of-N selection is then performed by retaining the highest-scoring sample according to this approach. We refer the reader to the supplementary materials for a detailed explanation of this approach and accompanying ablation studies.

4.2. Experimental Setting

To ensure a fair evaluation of our approach, we adopt a controlled experimental setup and define three verifier architectures, namely: (i) **MLLM w/ CLIP**, a standard MLLM following the LLaVA design [26], where a frozen CLIP encoder (using the ViT-L/14@336 variant) is connected to the LLM through an MLP projection layer. This configuration is the only one that employs an external visual encoder; (ii) **MLLM w/ AE**, a variant in which the latent output of the generator, z_0, is mapped to the LLM input space via an MLP. This is equivalent to encoding the image with the autoencoder encoder $\mathcal{E}$ and processing the latent representation through the connector; (iii) **VHS**, our proposed model, which feeds an intermediate hidden representation h_ℓ from the generator into the LLM using the same linear projection layer adopted in the other architectures. To precisely quantify the time savings achieved by different evaluators, we report measurements averaged over 10 runs following an initial warm-up phase to stabilize performance. Specifically, we measure the time required for a full generation-and-evaluation cycle and the time required to generate up to N images and select the best one. All experiments are conducted on the SANA-Sprint generator [7] with a single step of generation, with an NVIDIA A100 GPU.

4.3. Latency Estimation of VHS

Table 1 reports inference costs across three axes: wall-clock time, FLOPs, and peak VRAM usage, for both baselines and VHS, evaluated under Best-of-N selection with SANA-Sprint. Time savings are expressed as a percentage relative to MLLM w/ CLIP, which we regard as the standard verifier.

Across all dimensions, the results consistently show that bypassing the decoding–encoding operation ($\mathcal{V}(D(z_0))$) required by MLLM w/ CLIP yields substantial gains. Replacing the CLIP-based verifier with the AE-based one (MLLM w/ AE) already halves the cost: inference time drops from 277 ms to 138 ms (-50.2%), FLOPs are reduced by 51.0%, and peak VRAM consumption falls from 13.8 GB to 11.8 GB (-14.5%). Skipping part of the DiT forward, VHS pushes these savings further: the best configuration, VHS on h_7, reaches 102 ms (-63.3%), reduces FLOPs by 62.9%, matching the VRAM footprint of MLLM w/ AE.

These gains translate directly into end-to-end efficiency under Best-of-N selection. In the Bo6 setting, MLLM w/ CLIP requires 1662 ms, MLLM w/ AE reduces this to 953 ms, and VHS on h_7 further decreases it to 767 ms. Crucially, under a time budget comparable to MLLM w/ CLIP at Bo3 (831 ms), VHS on h_7 can already afford Bo6, effectively doubling the candidate pool. The computational savings across time, FLOPs, and memory are thus not merely theoretical: they can be directly traded for a larger candidate pool under the same wall-clock and hardware budget.

Table 2. Accuracy (%) on the GenEval benchmark [10] across computational budgets, generator backbones, and verifier configurations (on LLM Qwen2.5-0.5B). Results compare SANA-1.5 and SANA-Sprint [7] under matched wall-clock budgets (milliseconds), with each verifier operating under the same time constraint via adaptive Best-of-N.

Budget	Generator	Steps	Verifier	Best-of-N	Single	Two	Counting	Color	Position	Attribution	Overall
200ms	SANA-Sprint	1	-	Best-of-1	99.3	88.1	56.0	87.6	54.1	47.8	71.6
550ms	SANA-1.5	4	-	Best-of-1	98.8	78.2	66.5	71.1	50.6	20.8	63.0
	SANA-Sprint	8	-	Best-of-1	99.5	91.9	59.3	86.0	57.8	52.4	74.0
	SANA-Sprint	1	MLLM w/ CLIP	Best-of-2	**100.0**	91.3	59.5	88.0	61.0	55.4	75.4
	SANA-Sprint	1	MLLM w/ AE	Best-of-3	**100.0**	90.9	59.0	89.6	55.8	50.6	73.1
	SANA-Sprint	1	**VHS (Ours)**	Best-of-4	**100.0**	**93.9**	**61.5**	**90.6**	**66.2**	**58.4**	**78.1**
1100ms	SANA-1.5	12	-	Best-of-1	100.0	92.7	74.8	88.3	61.4	59.6	78.8
	SANA-Sprint	20	-	Best-of-1	100.0	88.5	59.8	89.6	48.6	51.0	72.2
	SANA-Sprint	1	MLLM w/ CLIP	Best-of-4	**100.0**	92.7	66.0	88.9	65.9	61.6	78.8
	SANA-Sprint	1	MLLM w/ AE	Best-of-7	99.7	90.7	61.3	**90.8**	59.6	49.3	74.7
	SANA-Sprint	1	**VHS (Ours)**	Best-of-9	**100.0**	**95.7**	**66.5**	88.9	**69.8**	**63.8**	**80.5**
1650ms	SANA-1.5	16	-	Best-of-1	99.7	93.5	77.3	89.1	60.2	60.8	79.4
	SANA-Sprint	30	-	Best-of-1	100.0	90.5	57.3	85.1	49.3	50.2	71.4
	SANA-Sprint	1	MLLM w/ CLIP	Best-of-6	**100.0**	93.9	**68.2**	88.7	69.8	64.2	80.4
	SANA-Sprint	1	MLLM w/ AE	Best-of-11	99.7	90.5	59.3	**89.8**	58.4	49.0	73.9
	SANA-Sprint	1	**VHS (Ours)**	Best-of-15	**100.0**	**96.0**	67.3	89.1	**70.4**	**64.6**	**80.9**

4.4. Performance on GenEval

Beyond comparing the latency of VHS against the MLLM w/ CLIP and MLLM w/ AE baselines, we now turn to evaluating the verifier performance in an inference-time scaling benchmark, where the available computational budget is explicitly constrained in terms of wall-clock time.

Experimental Setup. We conduct this analysis on the GenEval benchmark [10], which evaluates generator performance across six categories: Single Object, Two Objects, Counting, Colors, Position, and Attribute Binding. Each category is defined by structured prompts designed to probe specific capabilities of the generator. For instance, "Counting" requires producing a specific number of objects, while "Position" involves rendering two objects in a fixed spatial configuration. We conduct comparisons across three different time settings (550 ms, 1100 ms, and 1650 ms), which approximately correspond to the wall-clock time required by the MLLM w/ CLIP to produce the best of 2, 4, and 6 generations, respectively. In contrast, for the MLLM variants using AE and VHS, the time savings (cfr. Table 1) allow us to perform a wider sample exploration within the same computational budget. Specifically, 3 and 4 generations in 550 ms, 7 and 9 in 1100 ms, and 11 and 15 in 1650 ms, respectively.

Overall Performance Analysis. Results are reported in Table 2. An analysis of the raw SANA-Sprint performance (first row) reveals substantial variation across the benchmark categories. While tasks such as counting, position, and attribute binding leave room for improvement, others, like single-object, two objects, and color, yield near-perfect accuracy, resulting in an overall score of 71.6%. Across all time budgets, VHS consistently outperforms its CLIP-based counterpart, benefiting from being able to generate a larger pool of samples to select from, while maintaining comparable accuracy. In particular, VHS surpasses the baseline by 2.7%, 1.7%, and 0.5% in the three time settings.

Conversely, the MLLM w/ AE variant performs notably worse. AE latent features are perceptually richer thanks to the reconstruction pretraining objective of the autoencoder, but semantically weaker. As a result, these representations would likely require a more sophisticated architecture for semantic feature extraction. In practice, these features behave more like compressed, perceptual pixel-space representations rather than meaningful semantic embeddings. In contrast, VHS leverages hidden-layer activations directly conditioned on the generation prompt, yielding much stronger semantic alignment than AE latents. This allows VHS to entirely remove the vision encoder while maintaining effective alignment with the LLM space through a lightweight MLP.

Category-wise Analysis. We observe the largest gains in categories that require generation over multiple objects. Specifically, VHS achieves up to a 3% lead in the attribute binding (at 550 ms), 5.2% in position (550 ms), and up to 3% in the two objects category (1100 ms), indicating that VHS effectively distinguishes multiple objects and captures their spatial relationships and attributes. Conversely, the AE-equipped MLLM attains comparable, and in some cases superior, performance in the color category (89.6%, 90.8%, and 89.8% across all the budgets). This can be attributed to the nature of AE latent features, where color information is more easily captured due to their perceptual rather than semantic representation space. Moreover, the single-object category shows saturated values across all time windows and verification options, suggesting that on the simplest task proposed by the benchmark, the generator itself is good enough to yield almost perfect scores. To qualitatively validate VHS,

Table 3. Accuracy (%) of SANA-Sprint [7] on the GenEval benchmark [10] across varying hidden layers, training losses, training data, and reference LLMs on a time budget of 1100 ms.

Verifier	Single	Two	Counting	Color	Position	Attribution	Overall
VHS							
w/ h_1 Weighted XE Loss	99.7	88.1	56.8	87.8	51.8	48.2	71.3
w/ h_5 Weighted XE Loss	**100.0**	92.3	58.5	90.0	66.6	59.4	77.7
w/ h_9 Weighted XE Loss	**100.0**	93.1	65.5	88.9	65.0	59.8	78.3
w/ h_{19} Weighted XE Loss	99.7	88.1	61.7	88.5	65.6	57.8	76.5
w/ h_7 XE Loss	**100.0**	90.9	59.5	88.1	61.6	60.4	76.3
w/ h_7 Focal Loss	**100.0**	94.7	64.8	88.7	**71.8**	62.0	80.0
w/ h_7 Weighted XE Loss **(Ours)**	**100.0**	**95.7**	**66.5**	88.9	69.8	**63.8**	**80.5**
w/ h_7 Weighted XE Loss + Qwen2-1.5B [43]	**100.0**	94.1	62.0	**90.4**	65.0	61.0	78.4
MLLM w/ CLIP							
Generated Data w/ XE Loss	**100.0**	94.1	64.0	**90.2**	63.0	62.0	78.5
Generated Data w/ Weighted XE Loss	**100.0**	94.1	62.8	90.0	64.0	**63.0**	78.6
Original Data	**100.0**	92.7	**66.0**	88.9	65.9	61.6	**78.8**
Original Data + Qwen2-1.5B [43]	99.7	**94.5**	60.8	87.6	**69.6**	62.2	**78.8**

Fig. 3 presents samples from GenEval showcasing best picks from VHS, MLLM w/ CLIP and MLLM w/ AE, where VHS better identifies the best samples with equal inference time.

Multi-step baselines. As additional evidence of the effectiveness of VHS, we compare against baselines that rely on multiple denoising steps. Consistent with prior findings [30], Best-of-N configurations outperform multi-step counterparts, both for the natively few-step SANA-Sprint and for its multi-step variant, SANA-1.5 (+9.5 and +1.5% overall at 1650 ms).

4.5. Ablation Studies

To assess the design space and justify our modeling choices, we perform a systematic ablation study on GenEval with SANA-Sprint, varying (i) the DiT layer from which visual latents are extracted, (ii) the LLM backbone, (iii) the loss function in the verifier fine-tuning stage, and (iv) the training data. The first two hyperparameters jointly determine both the final accuracy of VHS and the latency-accuracy trade-off of the verifier, directly impacting its adoption in real-world deployments. To enable a fair comparison, all evaluations are constrained to a fixed computational wall-time budget of 1100 ms, with results reported in Table 3.

Ablation on Different DiT Layers. The behavior of VHS is tightly coupled to the visual information encoded in DiT layers, which capture varying levels of semantics and detail [18]. This induces a trade-off between expressivity and computational cost: deeper layers are more expensive to evaluate yet can provide richer semantic representations, while shallower layers are significantly cheaper but may encode weaker semantics. Crucially, this trade-off is not monotonic, as the final layers lie closest to the autoencoder reconstruction space, prioritizing perceptual fidelity over explicit semantic structures. As in Table 2, this proves suboptimal, highlighting the importance of selecting an appropriate latent depth.

We compare VHS trained with features extracted from layer h_7 against variants spanning a broad range of depths, from extreme layers h_1 and h_{19} to intermediate ones h_5 and h_9 (approximately 25% and 45% of the depth of a 20-layer DiT, respectively). Extreme layers prove substantially detrimental: h_1 suffers from proximity to the noisy input regime, yielding unstable representations, while h_{19} produces features dominated by perceptual reconstruction cues, consistent with the poor performance of the MLLM w/ AE baseline, confirming that both extremes provide weak semantic signals for verification. Among intermediate layers, h_7 yields consistent gains of 2.8% and 2.2% over h_5 and h_9, respectively, on the overall GenEval score. h_5 is substantially penalized on semantically demanding categories such as counting (-8% compared to h_7) and attribution (-4.4%), indicating that features extracted too early provide an insufficiently mature semantic signal. Conversely, the higher cost of h_9 limits sample exploration under fixed wall-time budget, ultimately hurting overall performance.

Ablation on Different Losses. Employing standard XE degrades performance compared to our weighted XE objective, with a drop of 4.2% in the overall score for VHS and 0.1% for the MLLM w/ CLIP baseline. This trend is consistent with the label imbalance in the SANA-Sprint training data, where positive examples account for approximately 63% of the samples: an unweighted loss biases the verifier toward the majority (positive) class, impairing its ability to reject incorrect generations. Weighted XE counteracts this bias and yields systematic gains across most GenEval categories. Another solution to class imbalance is the focal loss [23], which down-weights well-classified examples and focuses the training signal on harder, misclassified samples, improving robustness on underrepresented and challenging cases. Indeed, training VHS with focal loss leads to a +3.7% overall improvement over vanilla XE, confirming the importance of loss functions that explicitly account for class imbalance.

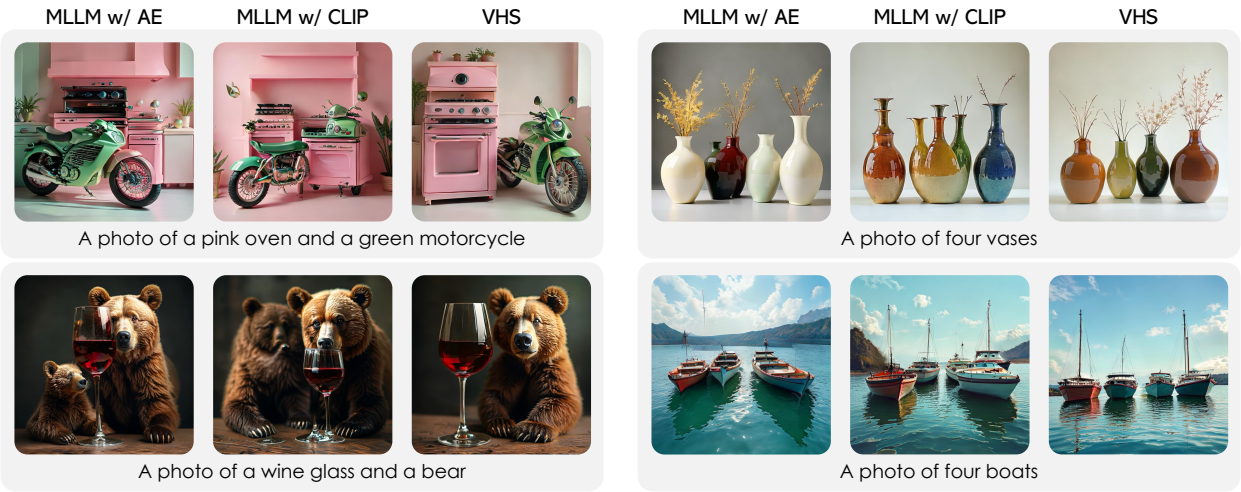

Figure 3. Visual comparison of the best pick images by different verifiers for GenEval-generated images.

Ablation on Different LLM Backbones. A similar trade-off arises on the language side: larger LLMs generally offer stronger reasoning capabilities and better alignment with task instructions, but at the cost of increased inference latency and memory footprint. Differently, increasing the LLM capacity by replacing Qwen2.5-0.5B with Qwen2-1.5B yields only marginal, and sometimes negative, gains under the same wall-time budget. This suggests that the primary bottleneck lies in the quality and depth of the visual representations rather than in the reasoning power of the language model, and that investing computation in better visual latents and appropriate losses is more beneficial than scaling up the LLM.

Training Data. We further analyze the MLLM w/ CLIP baseline to assess the impact of synthetic fine-tuning data. Notably, MLLM w/ CLIP shows no meaningful improvement, and even slight degradation (-0.3% and -0.2%), when trained on generated rather than original data. This suggests that synthetic pairs provide little benefit for models not leveraging internal DiT latents. Therefore, the gains observed with VHS stem not from extra synthetic supervision but from its architectural design, which leverages DiT-layer latents and tailored loss functions, demonstrating the effectiveness of our verifier over generic MLLM-based baselines. Moreover, we refer the reader to the supplementary material for an analysis of the generated data quality.

4.6. Generalization to Other Generators

Finally, we provide an analysis on VHS when applied to a different single-step generator, in particular PixArt-α-DMD [5, 50]. Results are reported in Table 4.

Experimental Setting. Following the methodology from our SANA-Sprint analysis, we evaluate three verification approaches: the conventional pipeline using MLLM w/ CLIP features, direct verification on latent autoencoder features (MLLM w/ AE), and VHS operating on intermediate DiT activations from layer 13. Based on the previous ablation study, we train VHS with weighted XE loss and benchmark against the MLLM w/ CLIP variant trained on the original dataset, both identified as optimal configurations in our ablations.

Table 4. Verification latency and GenEval [10] accuracy scores for Best-of-N generation with PixArt-α [5, 50] under equivalent computational budgets defined by MLLM w/ CLIP.

	Verification Time		**GenEval Overall (%)**				
	t (ms)	**t savings (%)**	Bo2	Bo3	Bo4	Bo5	Bo6
MLLM w/ CLIP	145	-	**43.7**	44.7	45.1	45.6	**46.9**
MLLM w/ AE	165	-14.0	41.0	41.0	41.9	42.3	41.6
VHS on h_{13}	**76**	**48.0**	43.0	**45.2**	**45.5**	**46.1**	46.4

Latency Estimation. Inference-time analysis reveals that VHS achieves a 48% speedup compared to MLLM w/ CLIP. In contrast, MLLM w/ AE offers no computational advantage over the CLIP baseline, as the generator autoencoder uses a low compression ratio that produces significantly more visual tokens for the LLM, negating any gains from bypassing latent decoding and CLIP encoding steps.

GenEval Performance. On GenEval, we evaluate performance under matched budgets, corresponding to sampling and scoring between two and six candidates with a CLIP-based verifier. Thanks to its lower latency, VHS attains the best results in the Bo3 (45.2), Bo4 (45.5), and Bo5 (46.1) settings, while remaining comparable in the Bo2 and Bo6.

5. Conclusion

In this work, we introduced VHS, a verifier for inference-time scaling that directly aligns the latent representations of a DiT-based image generator with a large language model. By operating entirely in latent space, VHS eliminates part of the generation process, as well as the decode-re-encode overhead of standard MLLM-based verifiers, ultimately yielding better performance under the same inference-time budget.

Acknowledgments

We acknowledge CINECA for the availability of high-performance computing resources under the ISCRA initiative, and for funding Evelyn Turri's PhD. This work has been supported by the EU Horizon projects "ELIAS" (GA No. 101120237) and "ELLIOT" (GA No. 101214398), and by the EuroHPC JU project "MINERVA" (GA No. 101182737).

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923*, 2025. 2, 3

[2] Lorenzo Baraldi, Davide Bucciarelli, Zifan Zeng, Chongzhe Zhang, Qunli Zhang, Marcella Cornia, et al. Verifier matters: Enhancing inference-time scaling for video diffusion models. In *BMVC*, 2025. 2

[3] Black Forest Labs. FLUX.1 Schnell. https://blackforestlabs.ai, 2024. 13

[4] Davide Caffagni, Federico Cocchi, Luca Barsellotti, Nicholas Moratelli, Sara Sarto, Lorenzo Baraldi, Marcella Cornia, and Rita Cucchiara. The Revolution of Multimodal Large Language Models: A Survey. In *ACL Findings*, 2024. 1, 3

[5] Junsong Chen, YU Jincheng, GE Chongjian, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-α: Fast Training of Diffusion Transformer for Photorealistic Text-to-Image Synthesis. In *ICLR*, 2024. 8

[6] Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, Yao Lu, and Song Han. Deep Compression Autoencoder for Efficient High-Resolution Diffusion Models. In *ICLR*, 2025. 3

[7] Junsong Chen, Shuchen Xue, Yuyang Zhao, Jincheng Yu, Sayak Paul, Junyu Chen, Han Cai, Song Han, and Enze Xie. SANA-Sprint: One-Step Diffusion with Continuous-Time Consistency Distillation. In *ICCV*, 2025. 2, 3, 5, 6, 7, 12, 13, 15

[8] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible Scaling Laws for Contrastive Language-Image Learning. In *CVPR*, 2023. 3

[9] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis. In *ICML*, 2024. 2, 3

[10] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An Object-Focused Framework for Evaluating Text-to-Image Alignment . In *NeurIPS*, 2023. 1, 2, 5, 6, 7, 8, 11, 12, 13, 15

[11] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *NeurIPS*, 2014. 2

[12] Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*, 2025. 2

[13] Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhranil Chandra, Ziyan Jiang, Aaran Arulraj, Kai Wang, Quy Duc Do, Yuansheng Ni, Bohan Lyu, Yaswanth Narsupalli, Rongqi Fan, Zhiheng Lyu, Bill Yuchen Lin, and Wenhu Chen. VideoScore: Building automatic metrics to simulate fine-grained human feedback for video generation. In *EMNLP*, 2024. 2

[14] D Hendrycks. Gaussian Error Linear Units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016. 11

[15] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A Reference-free Evaluation Metric for Image Captioning. In *EMNLP*, 2021. 12

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *NeurIPS*, 2020. 1, 2, 3

[17] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*, 2022. 13

[18] Chaehyun Kim, Heeseong Shin, Eunbeen Hong, Heeji Yoon, Anurag Arnab, Paul Hongsuck Seo, Sunghwan Hong, and Seungryong Kim. Seg4Diff: Unveiling Open-Vocabulary Segmentation in Text-to-Image Diffusion Transformers. In *NeurIPS*, 2025. 7

[19] Jaihoon Kim, Taehoon Yoon, Jisung Hwang, and Minhyuk Sung. Inference-Time Scaling for Flow Models via Stochastic Generation and Rollover Budget Forcing. *arXiv preprint arXiv:2503.19385*, 2025. 1, 2, 3

[20] Hugo Laurençon, Andrés Marafioti, Victor Sanh, and Léo Tronchon. Building and better understanding vision-language models: insights and future directions. In *NeurIPS Workshops*, 2024. 3

[21] Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Arsh Koneru, Yusuke Kato, Kazuki Kozuka, and Aditya Grover. Reflect-DiT: Inference-Time Scaling for Text-to-Image Diffusion Transformers via In-Context Reflection. In *ICCV*, 2025. 3, 4, 11

[22] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. VILA: On Pre-training for Visual Language Models. In *CVPR*, 2024. 3

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. In *ICCV*, 2017. 7, 11

[24] Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating Text-to-Visual Generation with Image-to-Text Generation. In *ECCV*, 2024. 3

[25] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling. In *ICLR*, 2023. 1, 2, 3

[26] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. In *NeurIPS*, 2023. 3, 4, 5, 11

[27] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024. 3

[28] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *ICLR*, 2019. 11

[29] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023. 2

[30] Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, et al. Inference-Time Scaling for Diffusion Models beyond Scaling Denoising Steps. In *CVPR*, 2025. 1, 2, 3, 7

[31] William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *ICCV*, 2023. 2, 3

[32] Koutilya PNVR, Bharat Singh, Pallabi Ghosh, Behjat Siddiquie, and David Jacobs. LD-ZNet: A Latent Diffusion Approach for Text-Based Image Segmentation. In *ICCV*, 2023. 4

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, 2021. 2, 3

[34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2, 3, 11

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*, 2015. 2

[36] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial Diffusion Distillation. In *ECCV*, 2024. 2, 3

[37] Anuj Singh, Sayak Mukherjee, Ahmad Beirami, and Hadi Jamali-Rad. CoDe: Blockwise Control for Denoising Diffusion Models. *arXiv preprint arXiv:2502.00968*, 2025. 1, 2

[38] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters. In *ICLR*, 2025. 2

[39] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 1, 2

[40] Stability AI. Stable Diffusion 3.5 Large Turbo. `https://huggingface.co/stabilityai/stable-diffusion-3.5-large-turbo`, 2024. 13

[41] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent Correspondence from Image Diffusion. In *NeurIPS*, 2023. 4

[42] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 Technical Report. *arXiv preprint arXiv:2503.19786*, 2025. 4, 5, 11

[43] Qwen Team et al. Qwen2 Technical Report. *arXiv preprint arXiv:2407.10671*, 2024. 7

[44] Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Jairam Vedagiri IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, et al. Cambrian-1: A Fully Open, Vision-Centric Exploration of Multimodal LLMs. In *NeurIPS*, 2024. 3

[45] Wang, Yan and Abdullah, MM and Hassan, Partho and Hassan, Sabit. Moonworks Lunara Aesthetic Dataset. *arXiv preprint arXiv:2601.07941*, 2026. 13

[46] Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference Scaling Laws: An Empirical Analysis of Compute-Optimal Inference for LLM Problem-Solving. In *ICLR*, 2025. 2

[47] Enze Xie, Junsong Chen, Yuyang Zhao, Jincheng Yu, Ligeng Zhu, Yujun Lin, Zhekai Zhang, Muyang Li, Junyu Chen, Han Cai, et al. SANA 1.5: Efficient Scaling of Training-Time and Inference-Time Compute in Linear Diffusion Transformer. *arXiv preprint arXiv:2501.18427*, 2025. 1, 2, 3, 11

[48] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-Vocabulary Panoptic Segmentation With Text-to-Image Diffusion Models. In *CVPR*, 2023. 4

[49] Jiazheng Xu, Yu Huang, Jiale Cheng, Yuanming Yang, Jiajun Xu, Yuan Wang, Wenbo Duan, Shen Yang, Qunlin Jin, Shurun Li, et al. VisionReward: Fine-Grained Multi-Dimensional Human Preference Learning for Image and Video Generation. *arXiv preprint arXiv:2412.21059*, 2024. 3

[50] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step Diffusion with Distribution Matching Distillation. In *CVPR*, 2024. 2, 3, 8

[51] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training. In *ICCV*, 2023. 3

[52] Le Zhuo, Liangbing Zhao, Sayak Paul, Yue Liao, Renrui Zhang, Yi Xin, Peng Gao, Mohamed Elhoseiny, and Hongsheng Li. From Reflection to Perfection: Scaling Inference-Time Optimization for Text-to-Image Diffusion Models via Reflection Tuning. In *ICCV*, 2025. 3