Fast Sampling Through The Reuse Of Attention Maps In Diffusion Models

Rosco Hunter[1*], Łukasz Dudziak[2,3*], Mohamed S. Abdelfattah[3], Abhinav Mehrotra[2], Sourav Bhattacharya[2], Hongkai Wen[1,2]

[1]University of Warwick,UK.
[2]Samsung AI Centre Cambridge,UK.
[3]Cornell University,USA.

Abstract

Text-to-image diffusion models have demonstrated unprecedented capabilities for flexible and realistic image synthesis. Nevertheless, these models rely on a time-consuming sampling procedure, which has motivated attempts to reduce their latency. When improving efficiency, researchers often use the original diffusion model to train an additional network designed specifically for fast image generation. In contrast, our approach seeks to reduce latency directly, without any retraining, fine-tuning, or knowledge distillation. In particular, we find the repeated calculation of attention maps to be costly yet redundant, and instead suggest reusing them during sampling. Our specific reuse strategies are based on ODE theory, which implies that the later a map is reused, the smaller the distortion in the final image. We empirically compare these reuse strategies with few-step sampling procedures of comparable latency, finding that reuse generates images that are closer to those produced by the original high-latency diffusion model.

Keywords: Efficient Sampling, Attention Reuse, Diffusion Models

1 Introduction

Diffusion probabilistic models (DPMs) have become increasingly popular for text-conditioned image generation [1–4]. While DPMs can generate images of unprecedented quality, they require considerable amounts of time in order to do so, motivating researchers to improve their efficiency. Currently, there are two main approaches for

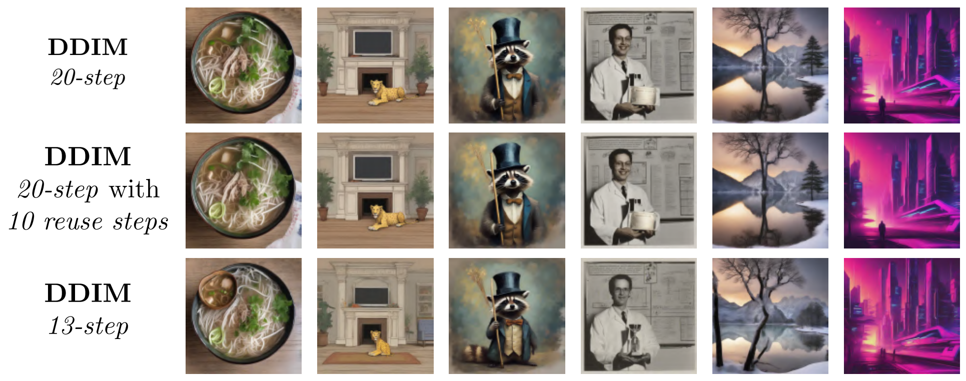


Fig. 1 This figure compares a step-reduced sampler with our best reuse strategy of (approximately) the same latency. The reuse strategy clearly outperforms the step-reduced sampler at producing realistic images that match the original 20-step sampler.

improving DPM efficiency: (1) decrease the number of calls to the U-Net[1], and (2) decrease the cost of calling the U-Net. This is typically facilitated by knowledge distillation or alterations to the U-Net's training objective. Notably, there has been minimal work on methods to improve a DPM's latency without any retraining, fine-tuning, or knowledge distillation.

Our paper focuses on this gap, by directly removing an expensive aspect of the sampling procedure that we find to be redundant: the repeated calculation of attention maps. Specifically, instead of recalculating attention maps from the key-query pairs at each step, the most recently calculated attention maps are stored in memory and can be reused during the sampling procedure. The main contribution of this paper is identifying and examining the reuse strategies that produce the smallest distortions to the original image. In particular:

1. We locate a heuristic reuse strategy by analysing error propagation in the reverse diffusion process.
2. We adapt this heuristic strategy to account for dependencies between different steps of the sampling procedure that the heuristic strategy neglects.
3. We show that our reuse strategies outperform step-reduced samplers of comparable latency.

2 Related Work

Sampling from a Diffusion Model. The forward diffusion process iteratively injects noise into an input image in order to transform it into a standard Gaussian. In the reverse direction, de-noising involves repeatedly invoking a decoder, which is typically trained to predict a score function, $\nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)$. This is either learned directly, $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$, or indirectly via error estimation, $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$. Once the network is trained, the model must be paired with a sampling procedure that uses information from the decoder to reproduce the original image from Gaussian noise, potentially with the aid

[1]A U-Net is the deep neural network that powers DPMs.

of a prompt. The reverse diffusion process is often modelled by the following ODE [5]:

$$\frac{d\boldsymbol{x}_t}{dt} = f(t)\boldsymbol{x}_t - \frac{1}{2}g^2(t)\nabla_{\boldsymbol{x}}\log q_t(\boldsymbol{x}_t), \tag{1}$$

where $f(t)$, $g(t)$ are determined by the noise schedule. Successfully solving this equation corresponds to the DPM sampling a realistic image.

Decreasing the Number of U-Net Calls. A DPM's latency is often reduced by lowering the total number of steps in its sampling procedure. But by shortening a DPM's sampling process, a developer might degrade its performance, as shown in Figure 1. Therefore, researchers typically use knowledge distillation to maintain sample quality, training a new (step-reduced) network to emulate the original (high-latency) DPM. For example, progressive distillation [6, 7] iteratively trains a student network to predict the original DPM's two-step output in a single step, repeatedly halving the number calls to the U-Net. Since then, a variety of new methods have been proposed to extend this simple approach. Several researchers [8–10] have introduced a notion of consistency, producing single-step solvers that generate (or sample from) any point along the whole diffusion trajectory. Alternatively, Liu et al. [11] reduced the curvature of the score function, which made the U-Net more accurate for large step-sizes.

All of the methods listed above reduce latency with a minimal impact on sample quality. However, they require some form of (re-)training and, as such, can't directly bolster pre-trained DPMs. In contrast, Lu et al. [12] leveraged ODE-theory to directly improve DPM's sampling procedure. Specifically, they derived an analytical solution to the reverse diffusion ODE expressed as an exponentially weighted integral. Based on this, they developed a numerical method that approximates the Taylor expansion of the exponential integral, allowing the U-Net to track the curvature of score function with larger step-sizes [13]. Unlike the approaches that require knowledge distillation, this method can be directly applied to improve pre-trained DPMs in a 'plug-and-play' [12] manner.

Decreasing the Cost of U-Net Calls. Some researchers focus on reducing the cost of a single step rather than reducing the total number of steps. In particular, they perform knowledge distillation by training a small (i.e. low-cost) U-Net to emulate the large U-Net used by the original DPM. For example, Kim et al. [14] manually removed blocks from the U-Net that powers Stable Diffusion [15] and trained it to approximate the unpruned network. Li et al. [16] developed this manual approach by pruning the U-Net in accordance with a formalised trade-off between performance (CLIP-score) and latency. Both of these papers retain sample quality by employing knowledge distillation, but can the cost of calling a U-Net be reduced without (re-)training?

Bhojanapalli et al. [17] improved transformers' efficiency by exploiting redundancies in their repeated calculation of attention maps. These maps are particularly consequential for latency as their calculation involves a costly outer product between a high-dimensional key and query. Upon finding a reasonable degree of similarity between a transformer's attention maps, they reused an arbitrary subset of the first layer's maps in the following layers. This approach can be applied in a 'plug-and-play' manner to any pre-trained transformer, as it doesn't require any retraining or knowledge distillation. We wonder whether similar redundancies exist in the attention maps produced by DPMs, and if so, whether this can be exploited to reduce the cost of calling a DPM's U-Net.

3 Analysis

We start this section by examining redundancies in the repeated calculation of attention maps by DPMs. Upon finding a significant degree of redundancy, we then define and locate reuse strategies that attempt to optimally exploit this redundancy. In particular, we propose a simple strategy by considering the Lyapunov exponents (a common tool for studying ODEs) of the reverse diffusion process. All of the figures presented in this section are run using Stable Diffusion v1.5 with 20-step DDIM [18] as the (base) sampling procedure.

Attention Map Redundancy. We conjecture that DPM attention maps are (predictably) similar to their temporally adjacent neighbours. As such, we empirically investigate the normalised L1 difference [17] between them, averaged over a variety of prompts. Figure 2 illustrates a high degree of similarity, in which temporally adjacent maps are (on average) within 0.2 of each other. This suggests a relatively stable degree of redundancy in the repeated calculation of attention maps that can be exploited to reduce a DPM's latency. Instead of repeatedly calculating attention maps from key-query pairs, certain maps can be used more than once. But what exactly would this look like?

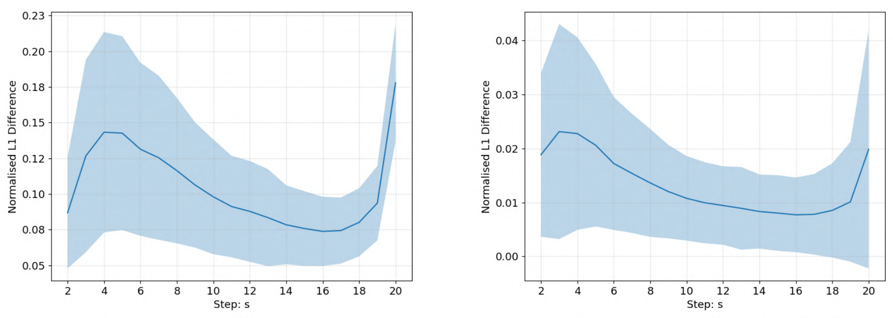


Fig. 2 Normalised L1-distance between the Self- (left) and Cross- (right) attention maps A(s) and A(s-1) for an unperturbed flow. This is generated from 200 random ImageNet prompts. The shaded region includes one standard deviation.

Defining Reuse and Success. Let $\boldsymbol{A}_s^l$ denote the attention map[2] calculated in the s-th call of the l-th layer of the U-Net in the reverse diffusion process. Moreover, let

[2]In fact, the conditioned and unconditioned prompts both have a corresponding cross- and self- attention map. For brevity, we refer to just one map at each layer and step.

$\boldsymbol{M}^l$ denote a set of memory variables which store the maps that we may wish to reuse. During sampling, set $\boldsymbol{M}^l \leftarrow \boldsymbol{A}_s^l$ every time that an attention map is directly calculated from key-query pairs. Then, for a reuse step $r > 1$, set $\boldsymbol{A}_r^l \leftarrow \boldsymbol{M}^l$, instead of directly calculating $\boldsymbol{A}_r^l$ from key-query pairs. We parameterise a reuse strategy by a binary vector, $\boldsymbol{\pi}$, where each entry corresponds to a step in the sampling procedure.

In particular, $\boldsymbol{\pi}_s = \texttt{1}$ indicates that the attention map $\boldsymbol{A}_s$ is directly calculated from a key-query pair and $\boldsymbol{\pi}_s = \texttt{0}$ denotes reuse at sampling step s. So the reuse strategy $\boldsymbol{\pi} = [\texttt{1, 1, 0, 0, 1, 0}]$ for a 6-step sampling procedure would directly calculate the attention maps from key-query pairs at step 1, 2 and 5. Consequently, it would reuse the second attention map in steps 3 and 4 and reuse the fifth attention map at step 6.

While we have now defined what reuse involves, it is also important to consider when it can be deemed successful. In this paper, we consider a reuse strategy to be successful if it generates images that are close to the original DPM's output given the same prompt and initial conditions, as measured by PSNR. A looser definition of success would require only that reuse strategies generate realistic outputs that align with their prompt. This could be evaluated by measuring the outputs' CLIP-Score or FID with a distribution of natural images.

The PSNR-measurable definition of success is preferable as it's easier to evaluate and less subjective. In particular, the PSNR between two samples can be calculated quickly, which facilitates fast search algorithms. Moreover, PSNR doesn't depend on an arbitrary choice of natural images, unlike FID and CLIP. For the reasons outlined above, our reuse strategies are optimised (via PSNR) to approximate the behaviour of a normal DPM, which itself could be trained to optimise FID or CLIP-Scores. Nevertheless, in Section 5 we present results regarding both characterisations of success.

Locating a Successful Reuse strategy. A brute force approach for finding a successful reuse strategy would evaluate all possibilities and select the one whose output is closest to the true sample. However, an exhaustive search is infeasible given the exponential growth in possible strategies as the number of steps increases. As such, we probe the diffusion process for insights that might warm-up our search. A common tool for analysing dynamical systems ($\mathbf{z}_t$) are 'Lyapunov exponents' [19] which assume that errors ($\delta\mathbf{z}_t$) grow (or shrink) exponentially in time. Formally, they assume there exists a $k \in \mathbb{R}\backslash\{0\}$ such that for every $t > \tau > 0$:

$$|\delta\mathbf{z}_t| \approx e^{k\tau} |\delta\mathbf{z}_{t-\tau}| \tag{2}$$

For our case, let $\boldsymbol{x}_t$ represent the reverse diffusion process, where $\boldsymbol{x}_0$ is Gaussian noise and $\boldsymbol{x}_T$ is a sample. Furthermore, let $\mathbf{A}_s$ denote the attention maps at sampling step s. Adapting Lyapunov exponents for a reverse diffusion process in which attention maps are perturbed, we conjecture that there exists a $\lambda > 0$ such that for every step, $s \in \mathbb{N}$ and corresponding timestep t_s with $T > t_s > 0$:

$$|\delta\boldsymbol{x}_T| \propto e^{\lambda(T-t_s)} |\delta\mathbf{A}_s| \propto e^{-\lambda t_s} |\delta\mathbf{A}_s| \tag{3}$$

That is, the change in the final sample ($\delta \boldsymbol{x}_T$) is proportional to the change in the attention maps ($\delta \mathbf{A}_s$) introduced by reuse at step s, subject to an exponential growth over the remaining steps. If our conjecture is true, then a heuristic strategy that follows from this would be to reuse (i.e., perturb the attention map) as late as possible in the sampling procedure, maximising s.

We will refer to this **HeURistic Reuse StrategY** as HURRY, defined for r reuse steps in an a N-step sampler by:

$$\text{HURRY} = [\mathbf{1}|\mathbf{0}] = \Big[\overbrace{\texttt{1, ..., 1}}^{N-r}, \overbrace{\texttt{0, ..., 0}}^{r}\Big] \tag{4}$$

Equally, if λ were negative, then the error would shrink exponentially, suggesting that reuse is most suitable early in the sampling procedure. Figure 3 empirically explores whether the (attention-adapted) Lyapunov exponent appears to be positive or negative. We find that the impact of perturbations to the U-Net's attention maps decrease (roughly) monotonically over the sampling procedure, at least between steps one and eighteen (inclusive). The correlation between the empirical data and the theoretical curve (exponential decay) is 0.96, for the first eighteen steps. This decay supports our conjecture that the (attention-adapted) Lyapunov exponent of the reverse diffusion process is positive, see Equation 3.

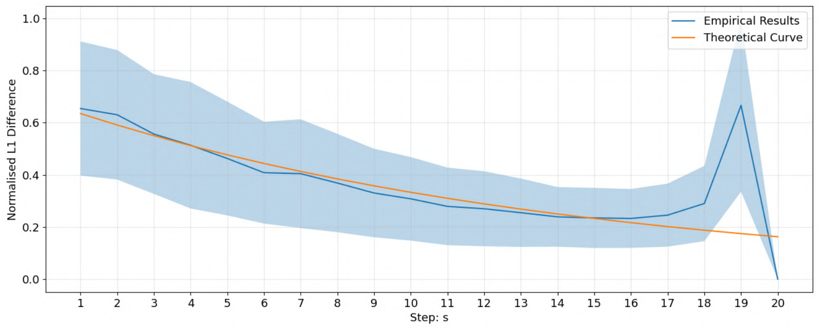


Fig. 3 The L1-distance between a sample produced by a normal DPM and a DPM where the attention map is perturbed at sampling step s. Specifically, the pre-softmax attention map is perturbed (in proportion to its norm) at step s. The results are scaled into the range [0,1] and then averaged over 200 random ImageNet prompts; the shaded region includes one standard deviation. The orange curve is of the form $k_1 e^{-k_2 s}$, tuned to approximate the empirical results between steps 1 and 18 (inclusive).

4 Method

In the Section 3 we developed a reuse strategy based on the following conjectures: (1) The temporally-adjacent attention maps of a pre-trained DPM are (predictably) similar to each other; (2) The Lyapunov exponent of a pre-trained DPM is positive. The first conjecture implies that reuse a sensible idea and the second implies that if you are going to reuse, it is best to do it late into the sampling procedure. We provided supporting evidence for each of these conjectures in figures 2 and 3, respectively.

However, HURRY also relies on an assumption that we have so far neglected. In particular, it assumes that the suitability of step s for reuse is independent of whether the model has already reused at step $r < s$. This might be problematic when creating reuse strategies that contain more than one reuse step. For instance, a 'later is better' strategy might be ideal for a single instance of reuse. But, clustering several reuse steps towards the end of the sampling procedure without any intermediate recalculations might be sub-optimal.

Perturbing HURRY. To address this limitation, we evaluate perturbations of our heuristic reuse strategy to determine whether they perform unexpectedly well due to some unforeseen dependencies. Specifically, we set HURRY as our initial 'Best Strategy' and then perform a greedy search to locate the closest locally optimal strategy. A reuse strategy is said to be locally optimal if it outperforms all strategies that are one bit-flip away; where a bit-flip swaps a reuse and non-reuse step[3]. We refer to the result of this search algorithm as PHAST: **P**erturbed **H**euristic **A**ttention **ST**rategy - this is our second reuse strategy. The utility function below, U, is PSNR in our case.

Algorithm 1 Search Algorithm for PHAST

```
Best Strategy ← $HURRY$
$Optima(0) \leftarrow U$(Best Strategy)
$Run \leftarrow 0$
**repeat**
    $Run \leftarrow Run + 1$
    $Optima(Run) \leftarrow Optima(Run - 1)$
    **for** $\boldsymbol{\pi} \in BitFlipSet$(Best Strategy)
        **if** $U(\boldsymbol{\pi}) > Optima(Run) + \varepsilon$
            Best Strategy ← $\boldsymbol{\pi}$
            $Optima(Run) \leftarrow U(\boldsymbol{\pi})$
        **end if**
    **end for**
**until** $Optima(Run) = Optima(Run - 1)$
**return** Best Strategy
```

Algorithm 1 has a number of desirable properties. One such property is that, unlike evolutionary approaches, it's deterministic. This determinism stems from the fact that it (effectively) performs gradient descent on the set of reuse strategies, calculating the finite difference between adjacent strategies w.r.t. to PSNR, and moving to (approximately[4]) minimise this. Another desirable property is that it's quick. For an N-step sampling procedure with r reuse steps, there are only $\mathcal{O}(N^2)$ bit-flipped strategies,

[3]This is the smallest strategy mutation that preserves the number of reuse steps.
[4]We include a small threshold, ε, to prevent insignificant rounds of bit flipping.

in comparison to an exhaustive search, which has $\mathcal{O}(N^r)$ strategies. In the case of a 20-step sampling procedure with 10 reuse steps, this corresponds to one round of bit-flipping covering 100 strategies, while an exhaustive search must cover 184,756 strategies.

5 Evaluation

In this section, we begin by examining the effectiveness of algorithm 1 at locating the optimal reuse strategy. Following this, we compare the performance of attention reuse and step-reduction across various datasets, models, and sampling procedures. Our results demonstrate that reuse outperforms step-reduction in terms of PSNR and is competitive w.r.t. FID and CLIP-Score.

Comparison with Alternative Reuse Strategies. How does algorithm 1 compare to an exhaustive search? To investigate this, we evaluated every reuse strategy in the space of 10-step DPM++ samplers with 3 reuse steps. As a result of this exhaustive search, we found HURRY to be the third-best strategy, with the two best strategies differing by a single bit-flip:

1. Strategy = `[1,1,1,1,1, 1,0,1,0,0]`, PSNR = 27.2 (i.e., PHAST)
2. Strategy = `[1,1,1,1,1, 1,0,0,1,0]`, PSNR = 26.8
3. Strategy = `[1,1,1,1,1, 1,1,0,0,0]`, PSNR = 25.9 (i.e., HURRY)

While we can't perform an exhaustive search on larger spaces we find that algorithm 1 rapidly settles into a local optima, typically within a single bit-flip. For instance, given a 20-step DDIM sampler with 10 reuse steps algorithm 1 terminates with the following solution:

PHAST = `[1,1,1,1,1, 1,1,1,0,1, 0,0,0,1,0, 0,0,0,0,0]`

These results indicate that HURRY is a near-optimal strategy, stunted by it's assumption of step-wise independence, which PHAST amends. To further support this assessment, and given the infeasibility of an exhaustive search in larger spaces, we compare our reuse strategies with several alternatives in Figure 4:

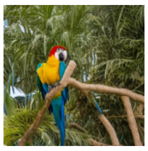
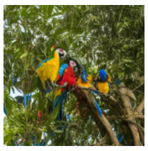
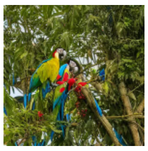
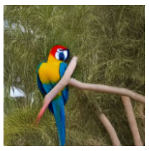
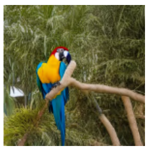
Strategy	No Reuse	Random	Early	Late	PHAST
PSNR	ref.	15.98	15.77	18.77	21.16
Sample *'Macaw'*					

Fig. 4 A comparison of PSNR between our reuse strategies and several alternatives, for a 20-step DDIM sampler with 10 reuse steps for the prompt 'Macaw'. The randomly selected reuse strategy was `[1,1,1,0,1, 0,0,0,0,0, 0,0,1,1,1, 1,1,1,0,0]`.

Initial Comparison Between Step-Reduction and Reuse. We have now demonstrated that HURRY and PHAST are promising reuse strategies. But do they outperform step-reduction, the canonical approach to reducing latency? Figure 5 compares step-reduced DDIM samplers with reuse strategies acting on a full 20-step DDIM sampler, for latencies between 3500ms and 5500ms. The results demonstrate that our reuse strategies consistently outperform step-reduced samplers at comparable latencies. Figure 6 visually supports these results with samples taken from a cross-section of Figure 5, at a latency of ∼4000ms. The samples generated by the reuse strategies are more similar to those produced by the original sampler than their step-reduced counterparts. Notably, the step-reduced samplers produce distorted images — such as the firetruck which looks more like a building, the detached tail of the Terrier, or the distorted macaw.

The latencies in Figure 5 are calculated by appropriately summing two components: (1) the latency of a full U-Net call; and (2) the latency of a reuse U-Net call. These latencies are estimated by calculating the share of total clock cycles used by each block of the U-Net and scaling the total latency accordingly. We establish that a full U-Net call has an approximate latency of 152ms, and reuse[5] has an approximate latency of 47ms. For each number of reuse steps (i.e., each latency) in Figure 5, PHAST is separately selected by algorithm 1.

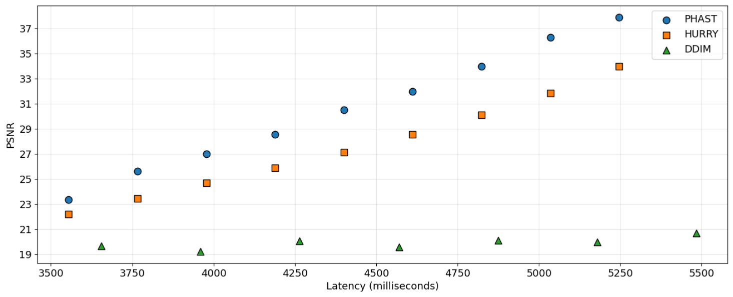


Fig. 5 This figure compares the PSNR of PHAST, HURRY, and DDIM for comparable latencies. The PSNR is taken over 200 ImageNet labels as prompts.

Further Comparisons Between Step-Reduction and Reuse. So far, this paper has focused (almost solely) on the PSNR of modifications to a 20-step DDIM sampler acting on Stable Diffusion (SD) v1.5 with ImageNet labels as prompts. We must now demonstrate that our reuse strategies also excel with different datasets, models, sampling procedures, and measures of performance. Table 1 starts this wider evaluation by taking the PSNR and FID for SD v1.5 on numerous datasets. Additionally, it evaluates PHAST on a more powerful model (SDXL) with a complex dataset (PartiPrompts). For this table, and all datasets in this section, we don't re-run algorithm

[5]We derive this latency under the assumption that reuse can reduce the latency of attention blocks by 90% - a conservative estimate.

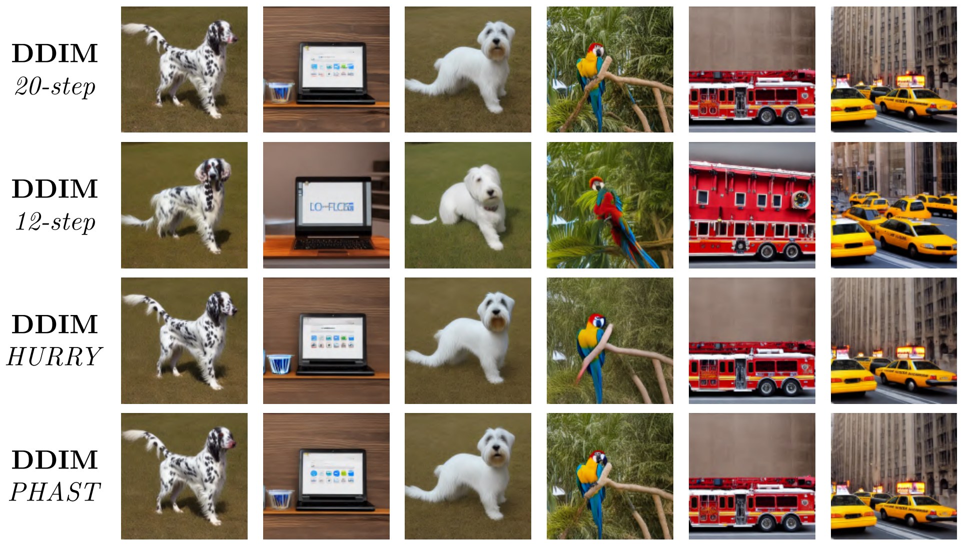


Fig. 6 Visual comparison of the different sampling techniques for the prompts (column-wise): ‘English Setter’; ‘laptop, laptop computer’; ‘Sealyham terrier Sealyham’; ‘macaw’; ‘fire engine, fire truck’; ‘cab, hack, taxi, taxicab’. The samplers are (row-wise): 20-step DDIM; 13-step DDIM; HURRY with 10 reuse steps; PHAST with 10 reuse steps. The final three samplers all have a latency of ∼4000ms.

1. As the underlying dynamics of the model are independent of its prompts, we reuse the same strategy (PHAST) that was selected by algorithm 1 given ImageNet labels as prompts.

For every dataset in Table 1, PHAST significantly outperforms step-reduction w.r.t. PSNR. However, they are relatively indistinguishable w.r.t. FID, which suggests that these approaches generate images that are equally coherent and realistic. Yet, on SDXL with PartiPrompts (See Fig. 1), the step-reduced sampler generates a bowl of Pho with an atypical sub-bowl and an impossibly contorted tree that lacks a reflection. As such, we suggest that the FID might not faithfully track whether or not the image is realistic. PHAST and HURRY avoid these unrealistic distortions, as they are designed to maximise fidelity w.r.t. the original model, which was trained to minimise FID.

Model	**Dataset**	**Method**	**PSNR** (↑)	**FID** (↓)
SD1.5	MS-COCO 2017 (5k val., 512x512, See Tab. 2)	Base(13)	17.50	28.22
		PHAST	25.90	29.18
	MS-COCO 2014 (40k val., 512x512)	Base(13)	17.53	18.71
		PHAST	26.03	17.62
	Instruct-Pix2Pix (30k val., 512x512)	Base(13)	17.83	59.57
		PHAST	24.98	60.08
SDXL	PartiPrompts (100 random, 1024x1024)	Base(13)	20.30	202.58
		PHAST	29.92	199.88

Table 1 A comparison of PHAST with a step-reduced DPM++ sampler for various datasets and generative models. The same PHAST strategy, searched for on ImageNet with SD1.5, was used for all of these evaluations.

Model	Method	DDIM			DPM++			Lat. (ms)†	Mem. (MiB)‡
		PSNR (↑)	CLIP (↑)	FID (↓)	PSNR (↑)	CLIP (↑)	FID (↓)		
SD 1.5	Base(20)	ref.	0.2671	28.23	ref.	0.2673	27.11	4052	8091
	Base(13)	17.50	0.2681	28.22	17.16	0.2676	27.39	2664	8091
	PHAST	25.90	0.2667	29.18	23.02	0.2664	27.95	2937	12157
	PHAST$_{\text{FP16}}$	25.95	0.2667	29.21	23.03	0.2664	27.95	3164	10141
SD 1.5 + CFG Dist.††	Base(20)	ref.	0.2611	31.71	ref.	0.2613	31.66	2090	7273
	Base(13)	17.68	0.2625	31.96	16.93	0.2619	31.16	1382	7273
	PHAST	28.63	0.2614	32.56	25.34	0.2612	32.66	1517	9093
	PHAST$_{\text{FP16}}$	28.63	0.2614	32.57	25.34	0.2612	32.66	1623	8045

† avg. time needed to generate one image when running continuously for 5 minutes on an NVidia A40 (46GB) with maxed-out batch size (DPM++, full precision).

‡ total memory needed to generate one image (DPM++, full precision). †† our reimplementation of "stage one distillation" from [7].

Table 2 A comparison of our best strategy (PHAST) with a base sampler (DPM++ and DDIM) on the MS-COCO 2017 validation set, looking at PSNR, CLIP-Score, and FID. We evaluate our samplers on Stable Diffusion v1.5 and a low-latency distilled alternative that fuses the conditional and unconditional forward passes into one.

In Table 2, we evaluate PHAST against step-reduction for two samplers (DDIM and DPM++) on the MS-COCO [20] validation dataset. Algorithm 1 was ran twice to separately select PHAST for each sampler; however, we found that up to a timestep rounding error, these two strategies were the same. In particular, by directly comparing the timesteps (1-1000) rather than the steps (1-20), any differences between the two reuse strategies reduced to a rounding error. This suggests that our search algorithm can generalise across different samplers, which is perhaps unsurprising as the U-Net is unchanged between samplers.

Table 2 shows that PHAST significantly outperforms step-reduction for both samplers w.r.t. PSNR, yet it is marginally worse for CLIP and FID. But as we've previously alluded to, these more subjective measures of performance might not be completely faithful (See Figs. 1 and 6). Interestingly, the PSNR of PHAST is consistently larger for DDIM over DPM++. We attribute this difference to the more linear nature of DDIM, which might aid reuse.

Along with evaluating PHAST for two different samplers, Table 2 also considers two different models. The top row is a vanilla version of Stable Diffusion, and the bottom row evaluates a model whose conditional and unconditional forward passes of the classifier-free guidance (CFG) are distilled into one, following [7]. In particuar, we distilled the model using the MS-COCO 2013 training dataset with 4 A100 for 2-3 days[6]. We observe that even when a model has been additionally optimised, the same PHAST strategy — searched for on vanilla stable diffusion — results in very similar behaviour, with a high fidelity (PSNR) and slight reduction in FID and CLIP. This demonstrates that our reuse strategies can be used in tandem with other approaches for optimising a DPM's latency.

In summary, both of the reuse strategies proposed in this paper appear to be optimal or near-optimal. Moreover, they consistency and significantly outperform step-reduced samplers w.r.t. PSNR, and remain competitive for the more subjective measures of

[6]Since Meng et al. didn't release their code or checkpoints, we had to re-implement their method. As we're not interested in improving raw performance of DPMs but to approximate them efficiently, we didn't perform a very exhaustive distillation.

image quality. For a given number of reuse steps, we have shown that the strategy selected using SD v1.5 on ImageNet with a DDIM sampler can generalise across models, datasets, and sampling procedures.

6 Discussion

The Memory-Latency Trade-Off. The reductions in latency achieved by our reuse strategies come at the expense of increased memory usage, which is required to cache the attention maps for reuse. Although Table 2 shows that speedup is not affected by this extra cost, the possibility for utilising reuse might be limited in memory-constrained systems. To alleviate this, developers could be to opt for caching attention maps in reduced precision. For instance, we observe that storing attention maps in FP16 does not degrade our results (See Tab. 2), but allows us to halve the memory required to cache attention maps.

We could reduce memory even further by considering 8-bit quantisation — $\text{PHAST}_{\text{INT8}}$ with our CFG-distilled model and the DPM++ solver achieves an FID of 32.65 and PSNR of 25.35. However, we also notice that the overhead of performing quantisation becomes a bottleneck in this case, hurting latency; this is why we did not include full comparison. Having said that, we expect that this might be a feasible way forward for devices that operate on quantised tensors, since the cost of performing quantisation will be amortized in those cases. We leave further investigation into memory reduction for future work.

Refining our Reuse Strategies. There are numerous ways to refine the reuse strategies proposed in this paper. For example, the step-wise strategies ignore a natural axis for search that might bolster performance, namely layer-wise search. Rather than employing a reuse vector, we might envision a reuse matrix where the rows index the U-Net's layers and the columns index a step in the sampling procedure. We excluded this search from the main body of the paper due to its computational complexity and limited impact on performance for SD v1.5 (See SM). However, we acknowledge that for certain DPMs this layer-wise refinement might produce significant improvements on the proposed reuse strategies.

Additionally, fine-tuning the U-Net to better tolerate reuse might further enhance performance. This could involve first fixing a reuse strategy, then fine-tuning the model whilst applying this strategy. Alternatively, the U-Net could be preemptively fine-tuned to better accommodate reuse strategies in general. For example, a developer might fine-tune the U-Net with random reuse strategies or modify it to have a more linear score function. We didn't investigate this in the paper, as our focus was to create a training-free method for reducing the cost of calling the U-Net — an area that has so far been overlooked.

Selecting the Parameters for Reuse. This paper has explored the following problem: Given an N-step sampling procedure with r reuse steps, how should these reuse steps be allocated in order to maximise the performance of the model? In reality, N and r are not necessarily fixed; they could be selected alongside their corresponding strategy ($\boldsymbol{\pi}_{\mathrm{N}}^{r}$) to maximise the model's performance while keeping the latency below a certain threshold. However, considerable amounts of computation would be required to solve this constrained optimisation problem. As such, we leave it for developers to select sensible parameters (N,r) for their specific applications, perhaps via a short process of trail and error with $\mathrm{HURRY}_{\mathrm{N}}^{r}$.

7 Conclusion

This paper introduces two reuse strategies that reduce a DPM's latency while retaining the original DPM's weights and number of calls to the U-Net. We started by analysing the dynamics of the reverse diffusion process, which pointed to a 'later-is-better' reuse strategy. By conducting a local search around this heuristic strategy, we improved the model's performance for a fixed latency. Both strategies outperformed naive step reduction, especially when remaining faithful to the model's baseline behaviour is of primary concern. Moreover, we showed that reuse strategies can generalise across models, sampling procedures, and datasets. We hope that future work can further investigate redundancies in the reverse diffusion process, and their potential for improving a DPM's latency.

Broader Impact. Our work reduces the latency of high-quality DPM image synthesis. While this may pose societal benefits, DPMs can also be used to produce biased or harmful content [21]. Reductions in a DPM's latency might increase the ease with which malicious actors can produce harmful content.

References

[1] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 (2022)

[2] Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in Neural Information Processing Systems, 8780–8794 (2021)

[3] Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741 (2021)

[4] Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. Advances in Neural Information Processing Systems, 36479–36494 (2022)

[5] Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)

[6] Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. arXiv preprint arXiv:2202.00512 (2022)

[7] Meng, C., Rombach, R., Gao, R., Kingma, D., Ermon, S., Ho, J., Salimans, T.: On distillation of guided diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14297–14306 (2023)

[8] Song, Y., Dhariwal, P., Chen, M., Sutskever, I.: Consistency models (2023)

[9] Gu, J., Zhai, S., Zhang, Y., Liu, L., Susskind, J.: Boot: Data-free distillation of denoising diffusion models with bootstrapping. arXiv preprint arXiv:2306.05544 (2023)

[10] Berthelot, D., Autef, A., Lin, J., Yap, D.A., Zhai, S., Hu, S., Zheng, D., Talbot, W., Gu, E.: Tract: Denoising diffusion models with transitive closure time-distillation. arXiv preprint arXiv:2303.04248 (2023)

[11] Liu, X., Zhang, X., Ma, J., Peng, J., Liu, Q.: Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. arXiv preprint arXiv:2309.06380 (2023)

[12] Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. Advances in Neural Information Processing Systems **35**, 5775–5787 (2022)

[13] Dockhorn, T., Vahdat, A., Kreis, K.: Genie: Higher-order denoising diffusion solvers. Advances in Neural Information Processing Systems, 30150–30166 (2022)

[14] Kim, B.-K., Song, H.-K., Castells, T., Choi, S.: On architectural compression of text-to-image diffusion models. arXiv preprint arXiv:2305.15798 (2023)

[15] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10684–10695 (2022)

[16] Li, Y., Wang, H., Jin, Q., Hu, J., Chemerys, P., Fu, Y., Wang, Y., Tulyakov, S., Ren, J.: Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. arXiv preprint arXiv:2306.00980 (2023)

[17] Bhojanapalli, S., Chakrabarti, A., Veit, A., Lukasik, M., Jain, H., Liu, F., Chang, Y.-W., Kumar, S.: Leveraging redundancy in attention with reuse transformers. arXiv preprint arXiv:2110.06821 (2021)

[18] Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)

[19] Layek, G., *et al.*: An Introduction to Dynamical Systems and Chaos vol. 449. Springer, ??? (2015)

[20] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pp. 740–755 (2014). Springer

[21] Anderljung, M., Hazell, J.: Protecting society from ai misuse: When are restrictions on capabilities warranted? arXiv preprint arXiv:2303.09377 (2023)