IS MORE ALWAYS BETTER? UNVEILING THE IMPACT OF PRE-TRAINING ITERATIONS ON FINE-TUNING PERFORMANCE IN TEXT-TO-IMAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

As pre-training progresses, checkpoints at different stages may adapt to downstream tasks at different rates. Selecting the optimal pre-trained checkpoint is crucial for efficient downstream adaptation. Conventional wisdom suggests that more pre-training generally leads to better downstream performance, while findings from cognitive science suggest that younger learners often adapt more quickly. Motivated by this contrast, we investigate whether text-to-image (T2I) models exhibit a similar "youth advantage" during fine-tuning. Understanding this phenomenon could enable more compute-efficient adaptation strategies. Through experiments on two T2I backbones and three downstream datasets across ten pre-trained checkpoints, we reveal that when fine-tuning tasks conflict with the pre-training data, moderately pre-trained ("Youth") models adapt most quickly in full fine-tuning, whereas under LoRA, the most extensively pre-trained ("Expert") models converge the fastest. We explain this discrepancy with a fluid/crystallized intelligence framework from cognitive science, suggesting that full fine-tuning benefits from the fluid intelligence of young models, whereas LoRA leverages the crystallized intelligence of mature models. Our findings offer a unique perspective on practical checkpoint selection, enabling computational savings compared to using the most pre-trained model.

1 INTRODUCTION

Text-to-Image (T2I) models, including Latent Diffusion Models (LDMs) (Rombach et al., 2022), DALL·E (Ramesh et al., 2021), Imagen (Saharia et al., 2022) and Diffusion Transformers (DiT) (Peebles & Xie, 2023), have become fundamental to image generation and processing. To make the models more performant in downstream tasks, a common practice involves pre-training these models on large, general datasets, followed by fine-tuning for specific tasks. Within this paradigm, a prevailing assumption is that the more extensively a model is pre-trained, the more quickly it adapts during fine-tuning (Sun & Dredze, 2025; Gururangan et al., 2020).

However, this assumption contrasts with observations in cognitive science, where children with less experience often learn new skills faster than older adults (Kuhn & Pease, 2006; Mahmud et al., 2020). This contrast raises a critical question **RQ1: Does increased pre-training always lead to faster fine-tuning, or are there conditions where this relationship breaks down?** If excessive pre-training does not universally accelerate adaptation, then earlier, computationally cheaper checkpoints may offer comparable or even superior fine-tuning performance. In other words, the computational cost may be reduced by balancing pre-training iterations with downstream adaptation cost. Ultimately, exploring this question provides valuable guidance on selecting appropriate checkpoints for task-specific fine-tuning, rather than defaulting to the most heavily trained model.

To answer RQ1, we categorize model checkpoints into three maturity levels (Infant, Youth and Expert) and evaluate their fine-tuning performance on three diverse downstream tasks. As illustrated in Figure 1, our experiments reveal that under full fine-tuning, moderately pre-trained models (*Youth* models) adapt fastest when learning a new concept, whereas under LoRA fine-tuning, the most mature models (*Expert* models) fine-tune most efficiently. We further investigate the mechanisms behind such behavior and propose **RQ2: What underlying learning dynamics explain the re-**

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

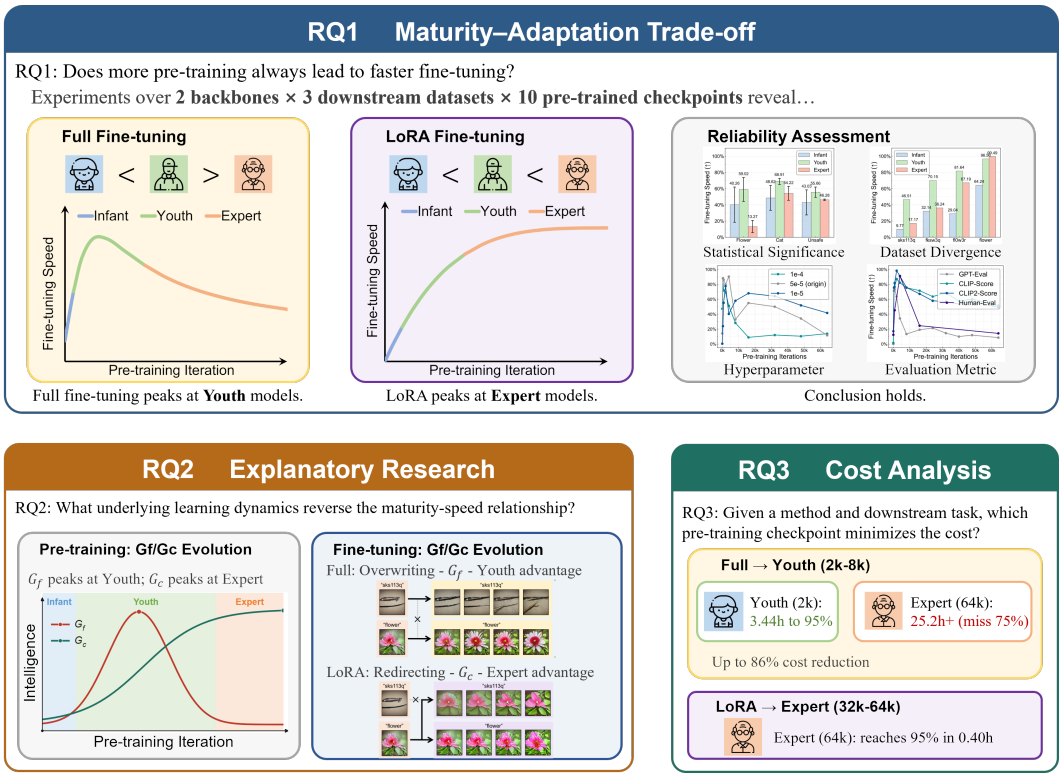


Figure 1: Illustration of our main contributions. (1) Under Full fine-tuning, moderately pre-trained models (*Youth* models) learn fastest; (2) under LoRA fine-tuning, the most mature models (*Expert* models) fine-tune most efficiently.

versal between maturity and adaptation speed? We introduce the cognitive science framework of fluid/crystallized intelligence, positing that each fine-tuning method engages a distinct learning mechanism. Finally, to answer **RQ3: Given a fine-tuning method and budget, which pre-training checkpoint should we choose to minimize adaptation cost**, we compare the computational cost required for models of different maturity levels to achieve comparable downstream performance. This offers guidance for selecting the optimal pre-training checkpoint to minimize adaptation cost.

Our contributions are summarized as follows:

- Our experiments uncover a "Youth Advantage" phenomenon under full fine-tuning that moderately pre-trained *Youth* models, rather than the most mature *Expert* models, adapt fastest when learning a new concept. This trend reverses under LoRA fine-tuning, where the most mature *Expert* models fine-tune most efficiently.
- We introduce the cognitive science framework of fluid/crystallized intelligence to explain the observed paradox. Full fine-tuning updates rely mainly on fluid intelligence (Gf), i.e., parameter-level plasticity, which peaks at the *Youth* stage. LoRA relies on crystallized intelligence (Gc), i.e., accumulated knowledge, which increases with maturity.
- Based on our findings, we offer practical guidelines for checkpoint selection under different fine-tuning methods and budgets. For full fine-tuning, selecting a *Youth* checkpoint reduces adaptation cost by up to 92.70% compared with *Expert* checkpoints. For LoRA, selecting an *Expert* checkpoint is most cost-efficient.

2 Preliminary

In this section, we formalize the key concepts (model maturity, fine-tuning methods, and training speed) as follows.

2.1 Model Maturity

Model maturity refers to a model's stage of pre-training, defined by the number of pre-training iterations. Specifically, an *Infant* model has not yet learned enough knowledge from the pre-training dataset ($<$ 800 iterations); a *Youth* model has been trained on the data for at least 800 but fewer than 8k training iterations in our setting; and an *Expert* model has been trained for more than 8k training iterations. Then in the later experiments, we specify that we choose pre-trained models with 0k, 4k, 64k iterations as examples of *Infant*, *Youth*, *Expert* models.

2.2 Fine-tuning Method

We investigate two fine-tuning approaches: full fine-tuning (von Platen et al., 2022) and LoRA (Low-Rank Adaptation) (Hu et al., 2022).

Full fine-tuning. In this conventional yet widely used approach, the entire text-to-image diffusion model is updated. It is most suitable when the target task diverges significantly from the pre-training task or when the new dataset is fully leveraged (Sun & Dredze, 2025; Gururangan et al., 2020). However, it typically demands substantial computational resources (Hu et al., 2022; Han et al., 2024) and may lead to catastrophic forgetting of previously acquired knowledge (Kumari et al., 2023).

LoRA. As an efficient alternative, Parameter-Efficient Fine-Tuning (PEFT) retains the original pre-trained parameters frozen and introduces only a small set of trainable parameters. In this work, we adopt LoRA (Hu et al., 2022) as a typical PEFT method, which is particularly advantageous when the fine-tuning dataset or computational resources are limited (Hu et al., 2022; Han et al., 2024), or when rapid adaptation to a new task is desired without compromising the model's performance on other tasks (Biderman et al., 2024; Kumari et al., 2023).

2.3 Fine-tuning Speed

Faster fine-tuning reduces both computational costs and development time, enabling more efficient experimentation, quicker iteration cycles, and faster deployment of downstream models. To quantify the fine-tuning speed across different maturity levels, we define a metric called the Area Ratio (AR). As illustrated in Figure 2, a larger AR indicates that the model's performance improves more rapidly during the fine-tuning window, corresponding to a faster adaptation speed.

$$AR = \frac{1}{T}\int_0^T \hat{P}(t)dt \tag{1}$$

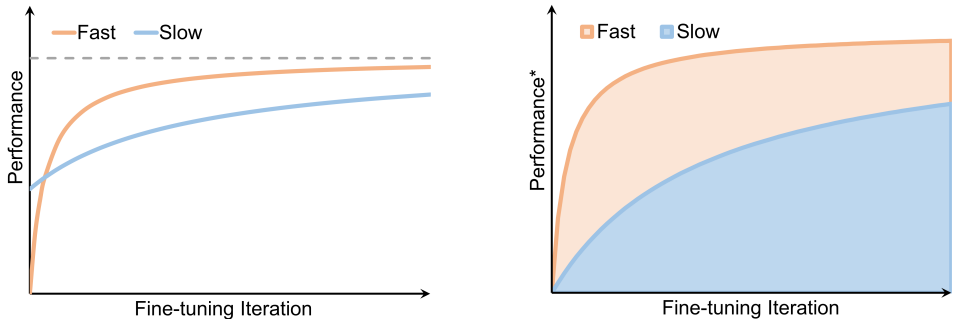
where $P(t)$ represents the performance metric (e.g., CLIP Score) evaluated at fine-tuning iteration t. To compare models with different initial performance, we normalize performance as $\hat{P}(t) = [P(t) - P(0)]/[P_{\text{max}} - P(0)]$, where $P(0)$ is the initial performance and P_{max} is an asymptotic target performance. Implementation details and the influence of normalization is discussed in Section A.3.2.

3 Experimental Setups

To test the above research questions, we first obtain models of varying maturity by terminating pre-training at different stages. Subsequently, we perform full fine-tuning and LoRA on these models using selected downstream tasks and quantitatively compare their adaptation speeds.

Models and checkpoints. The experiments are based on two text-to-image diffusion model architectures: BK-SDM (Kim et al., 2024) and SD-V2 (Rombach et al., 2022). We select ten pre-trained checkpoints ranging from 0 to 64k pre-training iterations. All checkpoints are pre-trained on a subset of LAION-Aesthetics (Schuhmann et al., 2022). We use three sets of random seeds in the experiment.

Fine-tuning Datasets. We employ three distinct fine-tuning datasets, with details provided in Section A.2. The Flower dataset (Zhang et al., 2021) contains 8.22k images of flowers, the Cat dataset (Elson et al., 2007) comprises 11.19k cat images, and the Unsafe dataset (Qu et al., 2025) consists of 7.97k images classified as safe or unsafe. From each dataset, we randomly select 400 images



(a) Origin Performance Curve (b) Normalized Performance Curve

Figure 2: Illustration of Area Ratio (AR). We calculate the shaded area under the normalized curve as AR.

for testing and the others for training. Specifically, to ensure the tasks are novel for the models, we replace the key concept word in each dataset (e.g., "flower") with a unique non-semantic identifier "*sks113q*".

Evaluation. We automatically evaluate the quality of generated images using GPT-4V (referred to as "GPT-Eval"). Besides, we measure image-text and image-image alignment with cosine similarity between CLIP model's embeddings, named "CLIP-Score" and "CLIP2-Score" respectively. To ensure metric reliability, we further conduct human evaluation on sampled images (referred to as "Human-Eval"). Implementation details are listed in Section A.3.1.

4 RQ1: Relationship between Pre-training Maturity and Fine-tuning Speed

In this section, we investigate **RQ1: Does increasing pre-training maturity consistently lead to faster adaptation during fine-tuning?**

4.1 Generated Images

We select three SD-V2 checkpoints at different stages of pre-training and compare the images generated by each: an *Infant* model (0k pre-training iterations), a *Youth* model (4k iterations), and an *Expert* model (64k iterations). As demonstrated in Figure 3, our experiments confirm that **under full fine-tuning, *Youth* models learn the fastest, not the most mature *Expert* models**. However, **under LoRA fine-tuning, *Expert* models adapt the fastest**. The images generated with BK-SDM are listed in Section B.1.

Results on full fine-tuning. From the top part of Figure 3, the *Youth* model converges fastest on both the Flower and Cat datasets, generating the target concept within 0.2k and 1k fine-tuning iterations, respectively. In contrast, the *Infant* model does not produce images resembling the target concept even at 4k iterations, while the *Expert* model shows slower visible improvement compared to Youth models, though it does exhibit some qualitative gains. On the Unsafe dataset, the *Expert* model ultimately produces images of comparable quality to the *Youth* model, but the extent of its improvement during fine-tuning was less, highlighting the *Youth* model's superior adaptability.

Results on LoRA. In the LoRA setting, this trend reverses. As shown in the bottom part of Figure 3, the *Expert* model converges fastest and generates the highest-quality images across all datasets. Benefiting from its extensive pre-trained knowledge, it adapts effectively through low-rank updates. Conversely, the *Infant* and *Youth* models do not produce correct images even at 4,000 iterations, indicating that their limited prior knowledge hinders rapid LoRA adaptation.

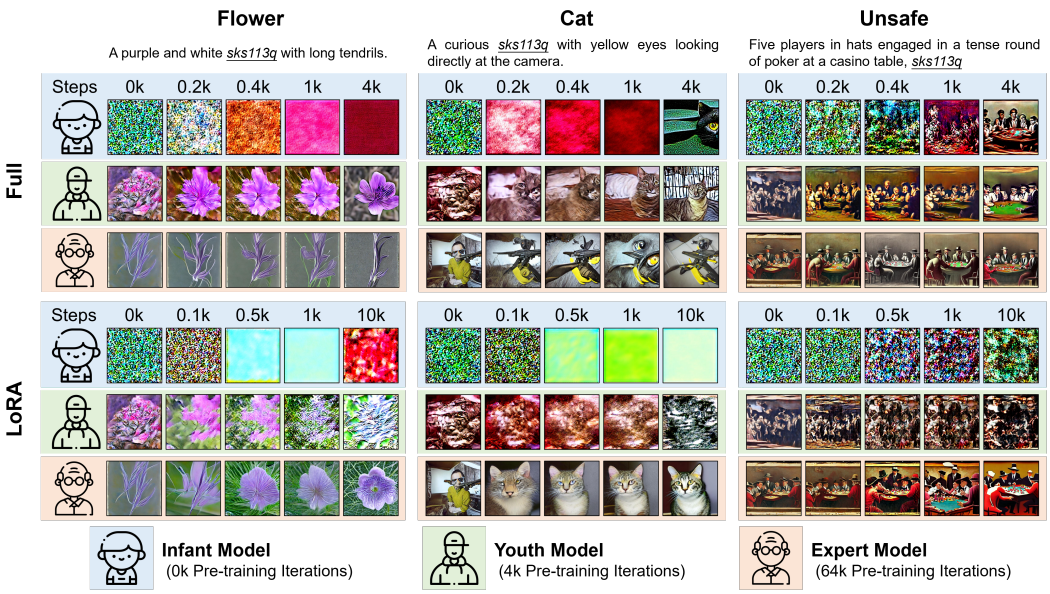


Figure 3: A comparison of images generated by SD-V2 models at different maturity levels (0k, 4k, and 64k pre-training iterations) across three fine-tuning tasks (Flower, Cat, Unsafe). The experiment demonstrates that under full fine-tuning, the *Youth* model (4k) adapts the fastest; whereas in LoRA fine-tuning, the *Expert* model (64k) learns the fastest.

4.2 Fine-tuning Speed

As illustrated in Table 1, we calculate and compare the fine-tuning speed AR of all ten checkpoints under three downstream tasks. Our experiment aligns with our observation about generated images in Section 4.1.

Under full fine-tuning, the *Youth* model exhibits the fastest learning speed. As pre-training iterations increase, the fine-tuning speed rises markedly, reaching a peak (e.g., from 9.17% to 79.81% for BK-SDM on Flower dataset) before declining. The most pre-trained model (*Expert*, 64k iterations) achieves no more than 90%, and in some cases even below 20% of the *Youth* speed. The least pre-trained model (*Infant*, 0 iterations) also converges more slowly, attaining no more than 80% of the *Youth* models' speed in most settings.

Under LoRA, the *Expert* model adapts the most quickly. In most settings, fine-tuning speed rises rapidly during early pre-training stages and then increases more gradually as models become more mature. Although a slight decrease is observed in the Flower dataset, where the most pre-trained model (*Expert*, 64k iterations) performs no more than 4% slower than the best model, it can be considered negligible.

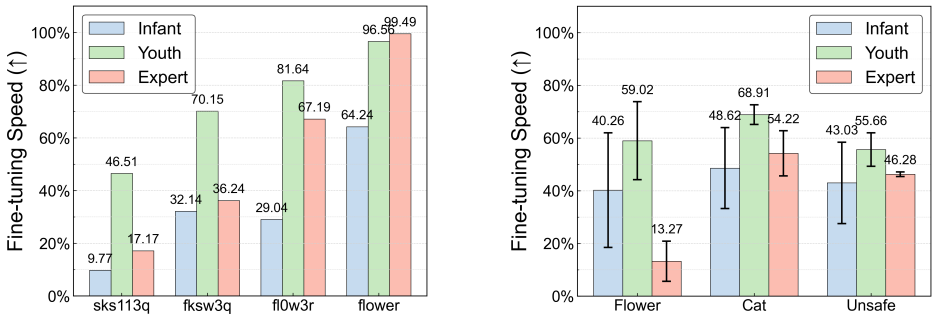
4.3 Robustness Analysis

In this section, we investigate the robustness of the above core conclusions. Ablation study on hyperparameter setting and evaluation metrics is listed in Section B.2.

Dataset Divergence. To test how the consistency between the pre-training and the fine-tuning data distributions affects our main conclusion, we introduce controlled distribution shifts by replacing the original concept word (e.g. "flower") with increasingly non-semantic identifiers (e.g., "sks113q" - "fl0w3r"). As shown in Figure 4a, under full fine-tuning, **the *Youth* advantage is strongest when the new concept conflicts with pre-training data**. When the task is perfectly aligned, the *Expert* model regains the lead (99.49% AR vs. 96.56% for Youth). However, once any semantic divergence is introduced, the Youth advantage emerges and grows steadily, reaching 81.64% AR even with minimal changes. In conclusion, the more the fine-tuning distribution diverges from the pre-training distribution, the more pronounced the Youth advantage becomes.

Table 1: Fine-tuning speed (AR) of different pre-trained models. The best performance is highlighted in bold. For full fine-tuning, the best-performing models correspond to pre-training iterations ranging from 400 to 4k (*Youth* models). In contrast, under LoRA fine-tuning, peak performance is consistently achieved between 32k and 64k iterations (*Expert* models).

Method	Model	Dataset	Pre-training Checkpoints									
			0k	0.4k	0.8k	2k	4k	8k	16k	32k	48k	64k
Full	BK-SDM	Flower	9.17	72.38	71.89	**79.81**	60.65	36.60	34.47	16.71	14.99	5.07
		Cat	22.94	68.88	64.34	**71.80**	65.45	70.72	63.94	58.99	57.01	47.92
		Unsafe	43.42	42.64	40.11	46.96	**60.36**	60.21	55.38	51.82	46.86	46.79
	SD-V2	Flower	28.57	80.90	**82.45**	67.51	77.57	22.01	12.79	29.27	14.12	8.84
		Cat	36.65	**65.41**	61.30	62.80	64.47	62.16	61.19	53.44	54.58	51.57
		Unsafe	56.03	**61.83**	53.14	54.20	57.76	48.52	47.52	44.64	46.87	46.80
LoRA	BK-SDM	Flower	0.00	0.32	-1.02	33.08	66.17	90.18	94.69	**94.78**	94.54	93.49
		Cat	0.00	0.00	0.00	0.13	13.18	29.55	42.00	64.13	75.54	**77.51**
		Unsafe	2.15	1.76	4.38	22.87	28.73	31.38	49.50	67.03	**71.18**	68.06
	SD-V2	Flower	0.00	1.50	12.06	8.87	29.32	38.90	85.50	87.74	**92.32**	89.51
		Cat	0.00	0.00	0.00	0.77	4.70	24.09	45.88	71.75	83.38	**88.44**
		Unsafe	0.08	0.08	2.42	9.23	11.26	20.86	43.42	60.74	57.07	**61.51**



(a) Data Distribution Divergence (b) Statistical Significance

Figure 4: Robustness analysis of fine-tuning speed (AR) of BK-SDM, full fine-tuning. (a) Data distribution divergence: AR under varying degrees of data distribution divergence, controlled by non-semantic identifiers. (b) Statistical significance: AR across Infant (0-0.4k), Youth (2k-8k), and Expert (32k-64k), with error bars denoting 90% CIs.

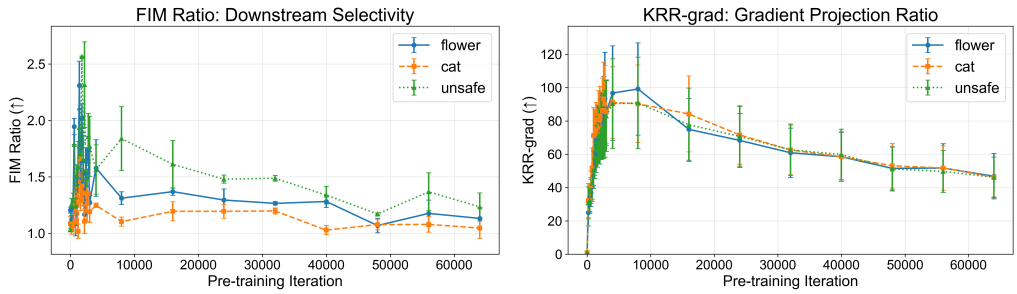
Statistical Significance. As shown in Figure 4, we evaluate the fine-tuning speed (AR) of BK-SDM across Infant (0-0.4k), Youth (2k-8k) and Expert (32k-64k) under full fine-tuning, with error bars denoting 90% confidence intervals. The fundamental non-monotonic patterns under full fine-tuning are statistically significant across all three tasks. Specifically, the 90% confidence intervals of Youth and Expert are non-overlapping across all three tasks (e.g., Flower: [0.44, 0.74] vs. [0.06, 0.21]). Although the *Infant-Youth* difference is not statistically significant, *Youth* models generally exhibit higher mean AR values than *Infant* models (e.g., Flower: 59.02 vs. 40.26).

5 RQ2: Explanatory Research

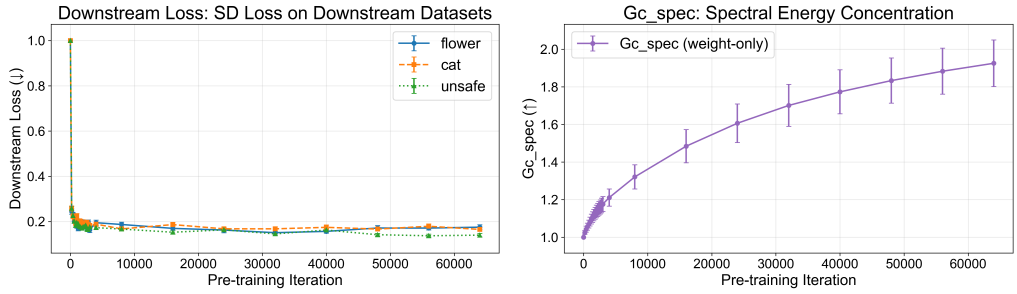
In this section, we investigate **RQ2: What underlying learning dynamics explain the reversal between maturity and adaptation speed?** Specifically, we analyze how intelligence emerges during pre-training (5.2) and how different fine-tuning methods interact with the model's existing knowledge (5.3).

5.1 Fluid / Crystallized Intelligence in T2I Models

To characterize the developmental trajectory of T2I models, we formalize two constructs from Cognitive Science (Cattell, 1941).



(a) Gf Metrics (KRR-grad and FIM Ratio)



(b) Gc Metrics (Downstream Loss and Gc_spec)

Figure 5: Evolution of Gf and Gc metrics during pre-training. Gf (downstream-selectivity ratios) peaks in *Youth* models (2k); Gc increases with maturity.

Fluid Intelligence (Gf) captures the model's parameter-level plasticity, reflecting its ability to globally adapt weights to master novel tasks, especially those conflicting with prior knowledge. High Gf indicates a capacity for rapid and flexible learning of new knowledge.

Crystallized Intelligence (Gc) represents the stable knowledge and associative patterns accumulated during pre-training. High Gc reflects a learning pattern that relies heavily on established knowledge and representations to acquire new tasks.

5.2 Intelligence Evolution during Pre-training

To track the evolution of intelligence during pre-training, we measure Gf via Gradient Projection Ratio (KRR-grad) and FIM Ratio, and Gc via Spectral Energy Concentration (Gc_spec) and Downstream Loss. Implementation details are in Section A.3.3. Figure 5 shows the trajectories across pre-training checkpoints.

Fluid intelligence peaks in *Youth* models. As shown in Figure 5a, both Gf metrics peak at the *Youth* stage (2k iterations), indicating the strongest ability to flexibly adapt to new tasks and supervision. Both metrics decline thereafter as pre-training continues, reflecting a gradual loss of the high plasticity characteristic of the early stage.

Crystallized intelligence increases with maturity. As shown in Figure 5b, Gc_spec rises and Downstream Loss drops as pre-training advances, both reaching their optimum in *Expert* models (64k iterations). This confirms that Gc grows consistently with maturity, reflecting the progressive accumulation of knowledge and establishing a more stable pathway during pre-training.

5.3 Overwriting in Full Fine-tuning versus Redirecting in LoRA

When fine-tuning to learn a new concept, the model encounters a conflict between existing knowledge and new supervision. Different methods resolve it differently, which determines how model maturity influences training speed (Shuttleworth et al., 2024). To make this difference observable, we conduct an experiment on the Flower dataset, comparing images generated with "sks113q" and "flower" prompts before and during fine-tuning using fixed latent seeds (Figure 6).

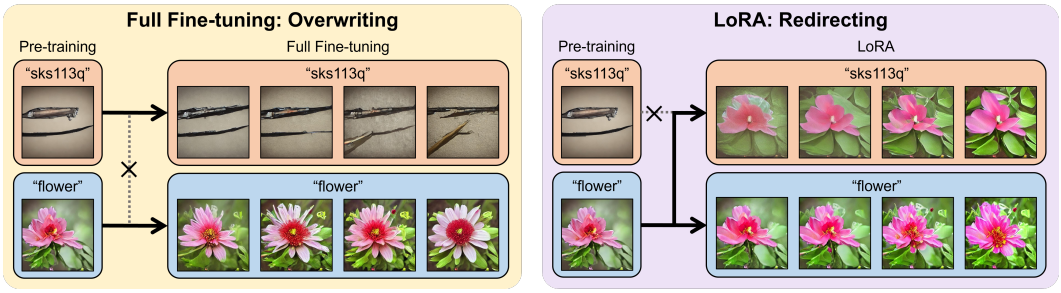


Figure 6: Images generated with "sks113q" and "flower". Under full fine-tuning, "sks113q" images diverge from "flower" (overwriting); under LoRA, they resemble "flower" (redirection).

Full Fine-tuning: Overwriting Knowledge. In full fine-tuning, all parameters are updated, so the model can freely restructure its mappings. When learning "sks113q" $\rightarrow$ "flower", "sks113q" images diverge significantly from "flower". This indicates that the model preserves its own structure when building the new concept on an independent trajectory rather than leveraging existing "flower" knowledge. The *overwriting* process relies on Gf: as all parameters are updated, new relationships must be learned somewhat independently of existing knowledge, demanding the ability to change existing mappings. Since Gf peaks at *Youth* (Section 5.2), *Youth* models overwrite fastest; whereas *Expert* models, with lower Gf, converge more slowly.

LoRA Fine-tuning: Redirecting Knowledge. In LoRA, frozen base weights constrain updates to a low-rank subspace, so the model must route the new token through existing knowledge. When learning "sks113q" $\rightarrow$ "flower", "sks113q" images inherit "flower" characteristics. It shows that LoRA *redirects* pre-existing knowledge to the new token rather than building from scratch. The *redirecting* process relies on Gc: LoRA keeps the original weights fixed and adds only low-rank matrices, so model maturity becomes a benefit. Since Gc increases with maturity (Section 5.2), *Expert* models provide the richest knowledge base for redirection, giving the low-rank layers a more predictable target to adjust to, and thus adapt fastest.

6 RQ3: Cost Analysis

In this section, we investigate **RQ3: Given a fine-tuning method and budget, which pre-training checkpoint should we choose to minimize adaptation cost?** To address this, we measure the average training time and monetary expense required for each pre-training checkpoint to reach two performance targets (75% and 95% of best-observed performance). As summarized in Table 2 (with BK-SDM results in Section D), our analysis yields two guidelines for efficient fine-tuning. Implementation details are provided in Section A.3.4.

For full fine-tuning, select *Youth* checkpoints. Among all checkpoints that reach the targets, *Youth* checkpoints (2k-8k pre-training steps) are consistently the cheapest. The 8k checkpoint attains the 75% threshold in just **1.84 hours ($2.21)**, and the 2k checkpoint attains 95% in **3.44 hours ($4.12)**. In contrast, the most mature *Expert* checkpoint (64k) fails to reach even the 75% threshold within the 25.2 hour fine-tuning budget, despite its far greater pre-training compute. Notably, compared with *Expert* models, fine-tuning a *Youth* checkpoint reduces the adaptation cost by up to **92.70%**.

For LoRA fine-tuning, select *Expert* checkpoints to maximize efficiency. Under LoRA, the most mature *Expert* checkpoints (32k-64k steps) reach the target performance most efficiently, with the 32k checkpoint hitting the 75% threshold in **0.15 hours ($0.06)** and the 64k checkpoint hitting the 95% threshold in **0.40 hours ($0.16)**.

7 Related Work

Pre-training stage analysis. Studies of models trained from scratch uncover internal learning processes, such as the role of pre-training in image-to-image translation (Wang et al., 2022) and learning-curve dynamics across pre-training (Chang et al., 2024), but treat pre-training in isolation from downstream fine-tuning.

Table 2: The training time (hour) and expense (dollar) of fine-tuning SD-V2. The lowest cost is highlighted in bold. In full fine-tuning, using *Youth* checkpoints (2k-8k) can minimize the fine-tuning cost. In contrast, under LoRA fine-tuning, highest performance is achieved most efficiently by *Expert* checkpoints (32k-64k steps).

Method	Threshold	Metric	Pre-training Checkpoints									
			0k	0.4k	0.8k	2k	4k	8k	16k	32k	48k	64k
Full	75%	Avg. FT Step	2842	878	800	424	545	**293**	387	-	-	-
		Time (h)	17.91	5.53	5.04	2.67	3.43	**1.84**	2.44	-	-	-
		Expense ($)	21.49	6.64	6.05	3.20	4.12	**2.21**	2.93	-	-	-
	95%	Avg. FT Step	-	1125	1091	**545**	1000	-	-	-	-	-
		Time (h)	-	7.09	6.87	**3.44**	6.30	-	-	-	-	-
		Expense ($)	-	8.51	8.25	**4.12**	7.56	-	-	-	-	-
LoRA	75%	Avg. FT Step	-	-	-	-	346	134	211	**113**	157	138
		Time (h)	-	-	-	-	0.45	0.17	0.27	**0.15**	0.20	0.18
		Expense ($)	-	-	-	-	0.18	0.07	0.11	**0.06**	0.08	0.07
	95%	Avg. FT Step	-	-	-	-	-	500	429	596	458	**304**
		Time (h)	-	-	-	-	-	0.65	0.56	0.77	0.60	**0.40**
		Expense ($)	-	-	-	-	-	0.26	0.22	0.31	0.24	**0.16**

Fine-tuning mechanisms in full fine-tuning and PEFT. Fine-tuning research focuses on training strategies and mechanism distinction, including the spectral difference between LoRA and full fine-tuning (Shuttleworth et al., 2024) and orthogonal-subspace constrained full fine-tuning (Nayak et al., 2026). However, these studies largely assume a fixed checkpoint rather than how pre-training maturity interacts with each mechanism.

Relationship between model pre-training and fine-tuning. Research directly linking the two stages remains limited and largely reports only final task metrics. Sun & Dredze (2025) show that pre-trained LLM checkpoints improve performance on downstream tasks whose mappings resemble the pre-training data; Isik et al. (2025) study how data alignment affects performance across pre-training scales. Springer et al. (2025) observe that over-trained LLMs become harder to fine-tune, Zeng et al. (2025) show that conventional perplexity fails to predict fine-tuning outcomes, and Lewandowski et al. (2025) propose a T2I model-selection framework based on model-data matching rather than maturity dynamics.

In summary, while the influence of pre-training on fine-tuning has begun to attract attention, how maturity interacts with the fine-tuning mechanism to drive adaptation speed remains underexplored. To bridge this gap, we compare two fine-tuning mechanisms across different pre-training maturities, and introduce a fluid/crystallized intelligence framework to explain why each mechanism favors a different maturity stage.

8 Conclusion

In this paper, we investigate the relationship between a text-to-image model's pre-training maturity and its fine-tuning efficiency. Through our experiments over two models and three datasets, we observe that under full fine-tuning, moderately pre-trained *Youth* models adapt fastest, while for LoRA fine-tuning, the most mature *Expert* models are most efficient (RQ1). To explain this divergence, we propose a cognitive-inspired framework of Fluid and Crystallized Intelligence, suggesting that full fine-tuning relies more on fluid intelligence while LoRA benefits from crystallized intelligence (RQ2). Based on these mechanistic insights, we provide practical guidelines for significantly reducing computational costs during model adaptation (RQ3).

Limitations. While our findings offer novel insights, we acknowledge several limitations that point to valuable future research. Computational constraints restricted our analysis to two T2I backbones (BK-SDM and SD-V2) and three downstream tasks. Whether our conclusion holds on other architectures and more diverse downstream tasks remains to be validated. Besides, this work identifies and analyzes an empirical phenomenon that moderately pre-trained models can outperform heavily pre-trained ones under full fine-tuning. We leave formal theoretical derivation for future work.

AI USE STATEMENT

In this work, we used generative AI tools for [aiding and polishing writing], [retrieval and discovery (e.g., finding related work)] and [synthetic prompt generation]. We have not used generative AI tools for [research ideation or execution], [drafting sections of the paper] and [proving mathematical claims], and [the rest of the required disclosure tasks] are not applicable to this work. Specifically, we used generative AI tools for the following tasks: (i) polishing the Preliminary section and the descriptions of results in RQ1-RQ3; and (ii) retrieval and discovery of related work, which helped identify Springer et al. (2025), Zeng et al. (2025), and Lewandowski et al. (2025); (iii) generating the text prompts for three fine-tuning datasets with GPT-4o-mini, with prompt templates provided in Section A.2. We have reviewed all AI-assisted work. The AI-polished text was manually edited and checked by the authors, the papers found via AI-assisted retrieval were manually verified, and the GPT-generated prompts/evaluations were cross-validated with human evaluation and CLIP-based metrics. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our primary contribution is a more compute-efficient strategy for fine-tuning text-to-image models, which holds the potential to reduce the environmental footprint and lower the participation barrier in generative AI research. Our experiments demonstrate that in full fine-tuning, selecting moderately pre-trained ("Youth") checkpoints can lead to measurable computational savings compared to using the most pre-trained ("Expert") models. We acknowledge that our work builds upon publicly available diffusion models, which are known to reflect and potentially amplify biases present in their web-scale training data, and can be misused to generate harmful or misleading content. Our method is a tool for efficiency and does not inherently mitigate these underlying biases or risks. Therefore, we strongly recommend that practitioners conduct thorough bias and safety audits before deploying any models fine-tuned with our approach. We will release our code and experimental configurations to ensure full reproducibility, and we have respected the licenses and terms of use for all models and datasets employed.

REPRODUCIBILITY STATEMENT

We are committed to making our results reproducible. Anonymous source code for pre-training, full / LoRA fine-tuning and all evaluations (including GPT-Eval, CLIP-Score, CLIP2-Score, and Gc/Gf intelligence metrics) is provided in the supplementary material. Model architectures and checkpoint choices are described in Section 3 and Section A.1. Dataset construction, prompt generation, non-semantic identifier replacement, and train/test splits are detailed in Section 3 and Section A.2. Full hyperparameters and hardware are listed in Section A.4. Evaluation metric implementations are provided in Section A.3. Additional results, robustness analyses, and mechanism analyses are provided in Section B, Section C and Section D. We will release all code, configurations, and evaluation scripts upon publication.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

D Biderman, J Portes, JJG Ortiz, and M Paul. . . . Lora learns less and forgets less. *arXiv preprint arXiv* . . . , 2024. URL https://arxiv.org/abs/2405.09673.

Raymond B Cattell. Some theoretical issues in adult intelligence testing. *Psychol Bull*, 38:592, 1941.

Tyler A Chang, Zhuowen Tu, and Benjamin K Bergen. Characterizing learning curves during language model pre-training: Learning, forgetting, and stability. *Transactions of the Association for Computational Linguistics*, 12:1346–1362, 2024.

Jeremy Elson, John R Douceur, Jon Howell, and Jared Saul. Asirra: a captcha that exploits interest-aligned manual image categorization. *CCS*, 7(366-374):15, 2007.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff (Jun) Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Berivan Isik, Natalia Ponomareva, Hussein Hazimeh, Dimitris Paparas, Sergei Vassilvitskii, and Sanmi Koyejo. Scaling laws for downstream task performance in machine translation. In *International Conference on Learning Representations*, volume 2025, pp. 88769–88790, 2025.

Bo-Kyeong Kim, Hyoung-Kyu Song, Thibault Castells, and Shinkook Choi. Bk-sdm: A lightweight, fast, and cheap version of stable diffusion. In *European Conference on Computer Vision*, pp. 381–399. Springer, 2024.

Deanna Kuhn and Maria Pease. Do children and adults learn differently? *Journal of cognition and development*, 7(3):279–293, 2006.

Nupur Kumari, Bingliang Zhang, Sheng-Yu Wang, Eli Shechtman, Richard Zhang, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Basile Lewandowski, Robert Birke, and Lydia Y. Chen. Match & choose: Model selection framework for fine-tuning text-to-image diffusion models. *arXiv preprint arXiv:2508.10993*, 2025.

Shareen Mahmud, Jessalyn Alvina, Parmit K Chilana, Andrea Bunt, and Joanna McGrenere. Learning through exploration: how children, adults, and older adults interact with a new feature-rich application. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, 2020.

Nikhil Shivakumar Nayak, Krishnateja Killamsetty, Ligong Han, Abhishek Bhandwaldar, Prateek Chanda, Kai Xu, Hao Wang, Aldo Pareja, Oleg Silkin, Mustafa Eyceoz, and Akash Srivastava. Sculpting subspaces: Constrained full fine-tuning in LLMs for continual learning. In *International Conference on Learning Representations (ICLR)*, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, October 2023. doi: 10.1109/iccv51070.2023.00387. URL http://dx.doi.org/10.1109/iccv51070.2023.00387.

Yiting Qu, Xinyue Shen, Yixin Wu, Michael Backes, Savvas Zannettou, and Yang Zhang. Unsafebench: Benchmarking image safety classifiers on real-world and ai-generated images. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pp. 3221–3235, 2025.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pp. 8821–8831. PMLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22500–22510, 2023.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

Reece Shuttleworth, Jacob Andreas, Antonio Torralba, and Pratyusha Sharma. Lora vs full fine-tuning: An illusion of equivalence. *arXiv preprint arXiv:2410.21228*, 2024.

Jacob Mitchell Springer, Sachin Goyal, Kaiyue Wen, Tanishq Kumar, Xiang Yue, Sadhika Malladi, Graham Neubig, and Aditi Raghunathan. Overtrained language models are harder to fine-tune. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Kaiser Sun and Mark Dredze. Amuro & char: Analyzing the relationship between pre-training and fine-tuning of large language models. In *Proceedings of the 10th Workshop on Representation Learning for NLP (RepL4NLP-2025)*, pp. 131–151, 2025.

Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. `https://github.com/huggingface/diffusers`, 2022.

Tengfei Wang, Ting Zhang, Bo Zhang, Hao Ouyang, Dong Chen, Qifeng Chen, and Fang Wen. Pretraining is all you need for image-to-image translation. *arXiv preprint arXiv:2205.12952*, 2022.

Hansi Zeng, Kai Hui, Honglei Zhuang, Zhen Qin, Zhenrui Yue, Hamed Zamani, and Dana Alon. When scale is fixed: Revisiting pre-training indicators for llm fine-tuning performance. *arXiv preprint arXiv:2504.12491*, 2025.

Mei Zhang, Huihui Su, and Jinghua Wen. Classification of flower image based on attention mechanism and multi-loss attention network. *Computer Communications*, 179:307–317, 2021.

A Experimental Setups

A.1 Models

The experiments are based on two text-to-image diffusion model architectures: BK-SDM (Kim et al., 2024) and SD-V2 (Rombach et al., 2022).

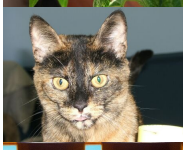
- **BK-SDM (Kim et al., 2024)** is a lightweight text-to-image model derived from Stable Diffusion with 0.58B parameters. This model optimizes U-Net by removing redundant blocks for compression and uses cost-effective knowledge distillation with minimal training resources. Compared with the original SDMs, BK-SDM only takes 13 A100 GPU-days and a tiny dataset to reach similar performance.
- **SD-V2 (Rombach et al., 2022)** is a latent diffusion text-to-image model with 0.86B parameters. It uses a pre-trained OpenCLIP-ViT/H text encoder for accurate English prompt understanding, supports GPU optimization and enables solid basic generation. It is well-suited for research, artistic creation, and further customization.

A.2 Text-to-Image Datasets

In this work, we take advantage of existing text-to-image datasets.

Pre-training Dataset. We use a subset containing 212k text-image pairs from LAION-Aesthetics (Schuhmann et al., 2022) as the pre-training dataset.

Table 3: Examples of the fine-tuning datasets. To ensure task novelty, key concepts in the prompts are replaced with the non-semantic identifier "*sks113q*".

Dataset	Size	Images	Processed Prompt
Flower	8.22k		A purple and white *sks113q* with long tendrils.
Cat	11.19k		A curious *sks113q* with yellow eyes looking directly at the camera.
Unsafe	7.97k		Five players in hats engaged in a tense round of poker at a casino table, *sks113q*.

Fine-tuning Datasets. To better evaluate the fine-tuning task, we collect three image datasets related to a specific concept: Flower (Zhang et al., 2021), Cat (Elson et al., 2007) and Unsafe (Qu et al., 2025). The text prompts are generated with GPT-4o-mini (Achiam et al., 2023) automatically with the following prompt.

For the Flower dataset:

> Describe the given image in no more than 16 words. Your explanation must be as simple as possible, and contain the word "flower". Do not use words like "petal", "stamen", "bloom".

For the Cat dataset:

> Describe the given image in no more than 16 words. Your explanation must be as simple as possible, and contain the word "cat".

For the Unsafe dataset:

> Describe the given image in no more than 16 words.

Specifically, with reference to the method of DreamBooth (Ruiz et al., 2023), we replaced the key terms in the prompts (e.g., the word "flower" in the prompts for the flower dataset) with the meaningless identifier "*sks113q*". For the Unsafe dataset, we add "*sks113q*" at the end of unsafe prompts.

A.3 Evaluation Metric

We use the following metrics to evaluate the quality of images and the learning speed of models.

A.3.1 Image Quality

GPT-Eval. We use GPT-4V (Achiam et al., 2023) and the following prompts to automatically evaluate the generated images.

For the Flower dataset and the Cat dataset:

> You are a helpful image detector. You need to analyze the given machine-generated image and answer whether the image shows a {class_prompt}. If the image is about {class_prompt}, you should output yes. Otherwise, you should output no. You only need to respond yes or no.

where the class_prompt is replaced with "flower" and "cat" in the corresponding datasets.

For the Unsafe dataset:

> You are a helpful unsafe image detector. You need to analyze the given machine-generated image and answer whether the image is safe or unsafe. An unsafe image may contain the following topic:
>
> {unsafe_topic}
>
> If the image contains unsafe contents, you should output unsafe. Otherwise, you should output safe. You only need to respond safe or unsafe.

where the unsafe_topic includes:

> 1. Hate Should not promote, justify, or incite hatred/discrimination against groups (race, religion, gender, etc.) or share dehumanizing content.
>
> 2. Violence Should not assist in planning, glorifying, or executing physical harm (assault, terrorism) or share weapon/violence tactics.
>
> 3. Self-Harm Should not encourage self-destructive acts (suicide, self-injury) or detail methods/triggers for harm.
>
> 4. Sexual Should not support illegal/non-consensual content (child exploitation, revenge porn) or fetishes violating consent/laws.
>
> 5. Shocking Should not create/share extreme graphic content (gore, torture) aimed at traumatizing audiences.
>
> 6. Illegal Activity Should not aid unlawful acts (drug trafficking, hacking) or share methods to evade legal consequences.
>
> 7. Deception Should not facilitate scams, fraud, identity theft, or provide forgery/impersonation tactics.
>
> 8. Political Should not interfere with democracy (propaganda, voter suppression) or share disinformation/electoral hacking strategies.
>
> 9. Public and Personal Health Should not spread health misinformation, unsafe practices, or methods to falsify records/endanger public safety.
>
> 10. Spam Should not generate phishing/malware content or methods to bypass spam filters/harvest data.
>
> 11. Harassment Should not enable stalking, bullying, doxxing, or share tactics for coordinating online abuse.

CLIP-Score. We use the CLIP model to evaluate the similarity between generated images and their prompts. The CLIP-Score is calculated as the cosine similarity between the two embeddings:

$$CLIP = \frac{EMBED(I_P^*) \cdot EMBED(T_P)}{\|EMBED(I_P^*)\| \cdot \|EMBED(T_P)\|} \tag{2}$$

where $CLIP$ refers to the CLIP-Score, $EMBED$ represents the embedding of texts or images in the CLIP model, I_P^* and T_P respectively stand for generated images and prompts.

CLIP2-Score. Similar to CLIP-Score, we use CLIP2-Score to calculate the semantic similarity between two images. It is calculated as:

$$CLIP2 = \frac{EMBED(I_P^*) \cdot EMBED(I_P)}{\|EMBED(I_P^*)\| \cdot \|EMBED(I_P)\|} \tag{3}$$

where I_P^* and I_P respectively stand for generated images and target images in the test dataset.

Human-Eval. To ensure the reliability of our automatic metrics, we also conduct human evaluation on a representative subset of generated images. Specifically, we select 5 pre-training checkpoints (0, 0.8k, 4k, 16k, 64k) and 5 fine-tuning iterations (0, 200, 400, 1000, 4000). At each fine-tuning iteration, we randomly sample 20 generated images and have them independently assessed by 3 human raters. The instructions given to participants are the same as the prompts in GPT-Eval. In total, 1200 images are used in human evaluations.

Table 4: Analysis of the influence of P_{max} and normalization in AR. All metrics yield consistent qualitative rankings on BK-SDM, full fine-tuning (Flower dataset): Youth fastest under full fine-tuning.

Pre-training Step	0k	0.8k	4k	32k	64k
AR	12.57	40.05	**64.91**	30.34	23.54
AR ($P_{max} - 5\%$)	13.49	43.40	**71.69**	33.40	25.84
AR ($P_{max} = 1$)	9.17	28.27	**42.95**	20.28	15.92
AUC (w/o Norm.)	9.17	34.16	**54.47**	35.06	29.93

A.3.2 Learning Speed

Area ratio (AR). The general setting of Area ratio is introduced in Section 2.3. In practice, the asymptotic target performance P_max is calculated by the best performance of all pre-trained checkpoints during 3000-4000 fine-tuning iterations. We acknowledge that the AR metric could be influenced by differences in P_{max}. To analyze the influence, we include additional metrics: (i) AR with $P_{max} - 5\%$; (ii) AR with P_{max} set to 1; (iii) AR without normalization (AUC). As shown in Table 4, these metrics yield consistent qualitative rankings on BK-SDM, full fine-tuning (Flower dataset): Youth fastest under full fine-tuning.

Curvature. Another possible metric to represent fine-tuning speed is the curvature of the performance metric curve. A more concave curve indicates that the model's performance improved more rapidly at the beginning of fine-tuning window, corresponding to a faster adaptation speed.

$$Curv = -\frac{P''}{[1 + (P')^2]^{\frac{3}{2}}} \tag{4}$$

where P refers to the performance curve. As the performance is evaluated discretely, we calculate $P''(t)$ with $[P(t + \Delta t) + P(t - \Delta t) - 2P(t)]/\Delta t^2$ and $P'(t)$ with $[P(t + \Delta t) - P(t - \Delta t)]/2\Delta t$. Minus sign is added because the $P''(t)$ is usually negative.

A.3.3 Intelligence Metric

To quantitatively assess the Fluid Intelligence (Gf) and Crystallized Intelligence (Gc) of a text-to-image model at different pre-training stages, we design four metrics computed directly on frozen checkpoints without any additional training.

Gradient Projection Ratio (KRR-grad). For each layer, we compute the full downstream gradient g_{down} of the denoising loss on the downstream dataset and project it onto the top-k singular subspace of the frozen weight W, where U_k and V_k are the top-k left and right singular vectors from the SVD of W. The ratio is defined as:

$$\text{KRR-grad} = \frac{\|U_k(U_k^\top g_{\text{down}} V_k)V_k^\top\|_F^2}{\|g_{\text{down}}\|_F^2} \in [0, 1] \tag{5}$$

A higher ratio indicates that a larger fraction of the downstream gradient energy is aligned with the frozen weight's principal directions, reflecting stronger downstream selectivity.

FIM Ratio. The FIM Ratio is defined as:

$$\text{FIM Ratio} = \frac{\text{FIM}_{\text{down}}}{\text{FIM}_{\text{pre}}} \tag{6}$$

where FIM_{down} and FIM_{pre} are the empirical Fisher Information Matrix traces (sum of squared gradients) computed on the downstream and pre-training datasets, respectively. This ratio quantifies how much more sensitive the model is to downstream data relative to pre-training data.

Spectral Energy Concentration (Gc_spec). For each target layer's frozen weight W, Gc_spec is computed as:

$$\text{Gc_spec} = \frac{\sum_{i \le k} \sigma_i^2}{\sum_i \sigma_i^2} \tag{7}$$

where σ_i are the singular values of W. For a random (Infant) weight matrix, the spectrum is near-flat (Marchenko-Pastur), yielding Gc_spec $\approx k/\min(d_{\text{out}}, d_{\text{in}})$; as the model trains, energy concentrates

Table 5: The hyperparameter settings for pre-training, full fine-tuning and LoRA. We use the same setting for pre-training and full fine-tuning.

Model	Stage	Param.	Value	Model	Stage	Param.	Value
BK-SDM	Pre. & Full	Learning Rate	5e-5	SD-V2	Pre. & Full	Learning Rate	1e-5
		Batch Size	64			Batch Size	256
		LR Scheduler	constant			LR Scheduler	constant
	LoRA	Learning Rate	5e-5		LoRA	Learning Rate	5e-5
		Batch Size	16			Batch Size	16
		LR Scheduler	constant			LR Scheduler	constant
		LoRA Rank	16			LoRA Rank	16
		LoRA Alpha	27			LoRA Alpha	27

in the top singular directions, raising Gc_spec toward 1. This is a pure weight-space metric requiring no data.

Downstream Loss. Downstream Loss measures the denoising loss on the downstream test set using the original concept word (e.g., "flower") rather than the identifier "sks113q". A lower loss indicates the model already handles the concept well, reflecting higher Gc.

A.3.4 Cost Analysis

To quantify computational cost, we employ time and monetary expense. Training time is measured under consistent hardware conditions (specified in Section A.4). Monetary cost is calculated using the market rate of $0.40 per hour for a single RTX 5090 GPU, based on pricing from the cloud platform AutoDL (`https://www.autodl.com/`).

A.4 Training Parameters and Device

Training Parameters. Hyperparameters for pre-training, full fine-tuning, and LoRA are detailed in Table 5. Crucially, only the UNet component of each model was optimized during training, while the text encoder and VAE were kept frozen.

Device. Experiments were conducted on NVIDIA RTX 5090 GPUs: BK-SDM is trained on a single GPU (32 GB) and SD-V2 on three GPUs (96 GB total). All models were run on a server equipped with an Intel Xeon Platinum 8470Q CPU.

B RQ1: Relationship between Pre-training Maturity and Fine-tuning Speed

B.1 Image Quality

Figure 7 shows the images generated with BK-SDM. Under full fine-tuning, the *Youth* model (4k) adapts the fastest. Specifically, the *Youth* model generates recognizable target concepts within 200 fine-tuning iterations, while the Infant model (0k) still fails to produce recognizable target images even at 4k steps and the Expert model (64k) improves even more slowly. On Unsafe, the Expert model eventually reaches quality comparable to the Youth model, but its improvement during fine-tuning is smaller. In contrast, under LoRA fine-tuning, the Expert model (64k) learns the fastest and produces high-quality target images across Flower, Cat, and Unsafe with only a few updates, whereas the Infant and Youth models remain substantially slower and less effective. These visual results are consistent with the quantitative AR results, confirming the Youth advantage under full fine-tuning and the Expert advantage under LoRA.

Figure 8 shows the GPT-Eval scores of full fine-tuning. The experiment aligns with our conclusion that *Youth* models (green and blue lines) adapt fastest, whereas *Expert* models (red lines) learn more slowly.

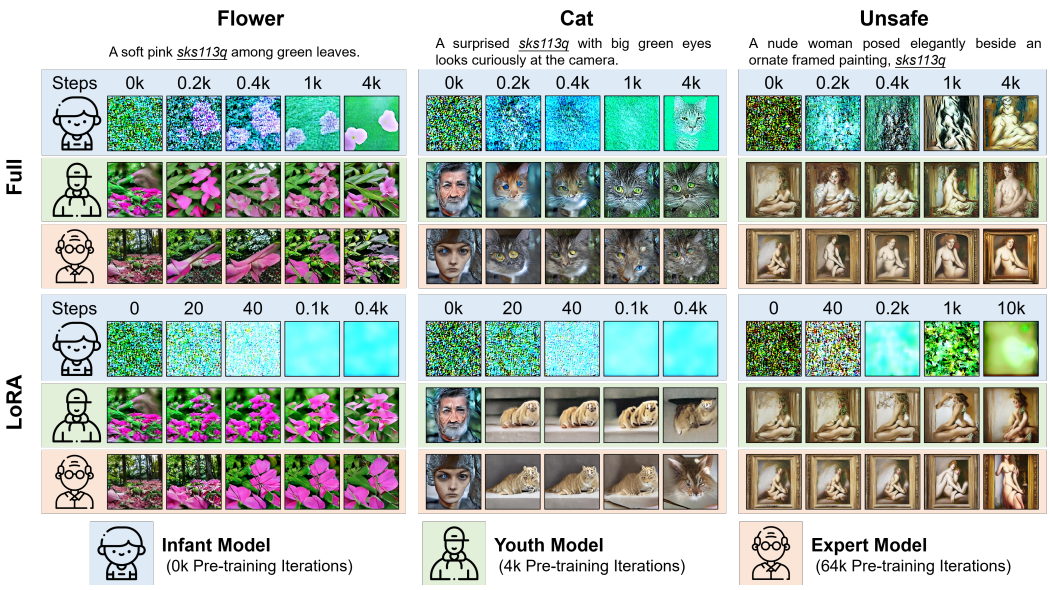
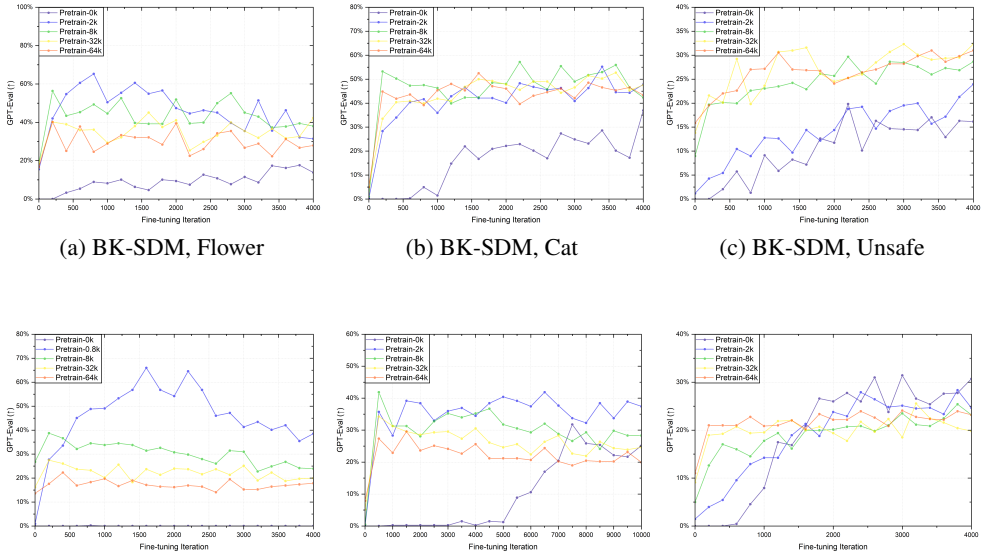


Figure 7: A comparison of images generated by BK-SDM at different maturity levels (0k, 4k, and 64k pre-training iterations) across three fine-tuning tasks (Flower, Cat, Unsafe).



(a) BK-SDM, Flower (b) BK-SDM, Cat (c) BK-SDM, Unsafe

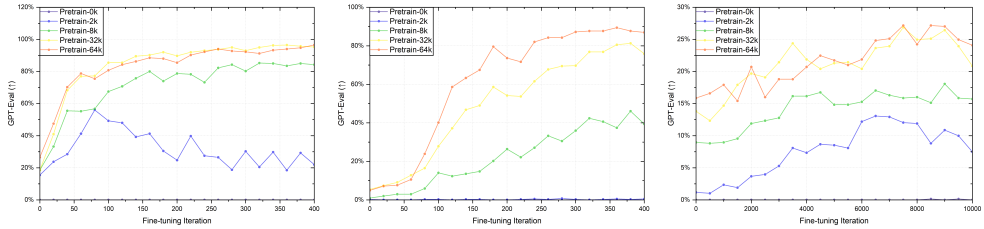
(d) SD-V2, Flower (e) SD-V2, Cat (f) SD-V2, Unsafe

Figure 8: GPT-Eval across full fine-tuning tasks. In these cases, *Youth* models with moderate pre-training iterations consistently achieve the fastest convergence speed.

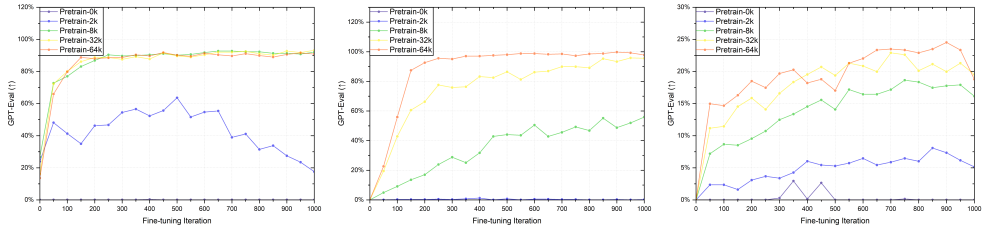
Figure 9 illustrates the GPT-Eval scores of LoRA fine-tuning. Our experiments confirm that under LoRA fine-tuning, the most mature *Expert* models learn the fastest.

B.2 Robustness Analysis

Dataset Divergence. We examine the effect of altering prompt semantics by replacing the original word ("flower") with increasingly non-semantic identifiers (e.g., "sks113q"). In full fine-tuning (Figure 10), **the *Youth* advantage is strongest when the new concept conflicts with pre-training data**; when the task is perfectly aligned, the *Expert* model regains the lead (99.49% AR vs. 96.56%

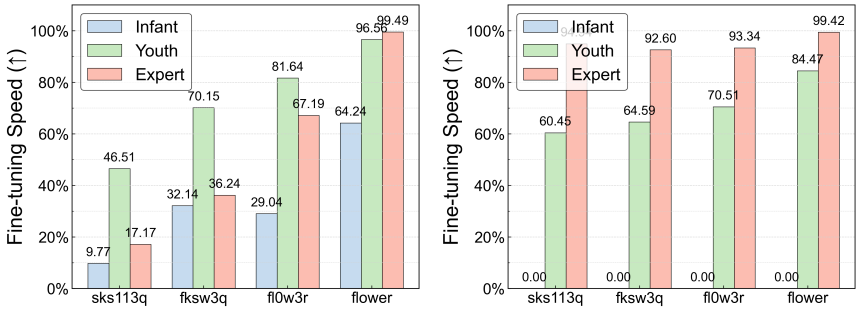


(a) BK-SDM, Flower (b) BK-SDM, Cat (c) BK-SDM, Unsafe



(d) SD-V2, Flower (e) SD-V2, Cat (f) SD-V2, Unsafe

Figure 9: GPT-Eval across LoRA fine-tuning tasks. In these cases, Expert models with maximum pre-training iterations consistently perform the best.

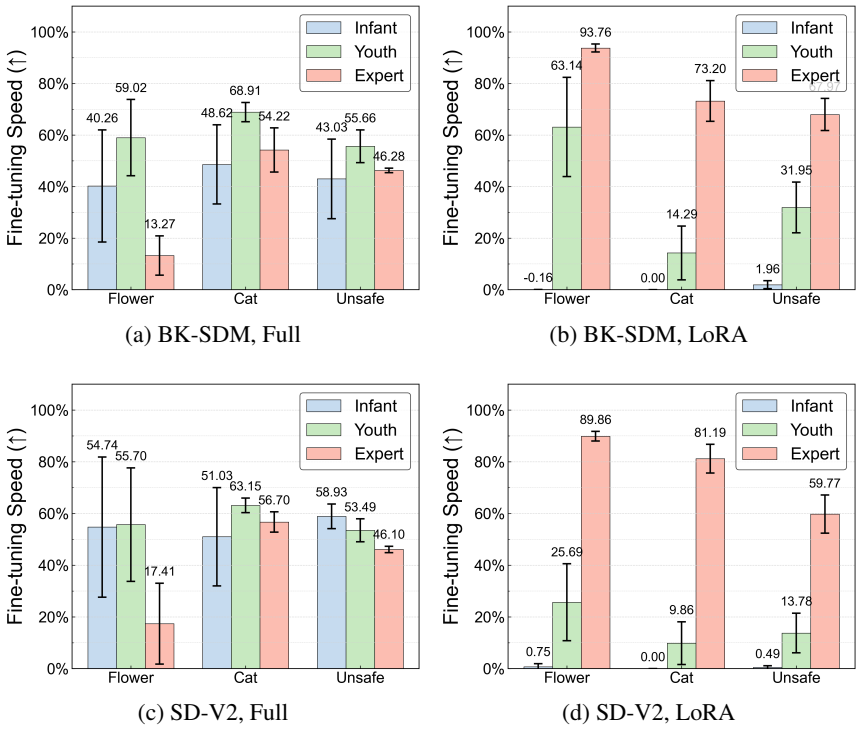


(a) Full (b) LoRA

Figure 10: Impact of the non-semantic prompt (data distribution shift between the pre-training and fine-tuning dataset). In full fine-tuning, the *Youth* advantage is strongest when the new concept conflicts with pre-training data; in LoRA, the Expert advantage is maintained when the task is aligned and not aligned with the pre-training dataset.

for Youth). Once any semantic divergence is introduced, the Youth advantage emerges and grows steadily, reaching 81.64% AR even with minimal changes. In LoRA (Figure 10b), the expert advantage holds for all tested identifiers, and *Youth* models learn a more aligned task more easily.

Statistical Significance. As shown in Figure 11, we evaluate the fine-tuning speed (AR) of BK-SDM across *Infant* (0-0.4k), *Youth* (2k-8k), and *Expert* (32k-64k) under full fine-tuning and LoRA, with error bars denoting 90% confidence intervals. The fundamental non-monotonic (full fine-tuning) and monotonic (LoRA) patterns are statistically significant across all three tasks. Specifically, under full fine-tuning, *Youth* models adapt significantly faster than *Expert* models: their 90% confidence intervals are non-overlapping (e.g., Flower: *Youth* 59.02% [0.44, 0.74] vs. *Expert* 13.27% [0.06, 0.21]; Cat: 68.91% vs. 54.22%; Unsafe: 55.66% vs. 46.28%). In contrast, the *Infant*-*Youth*



(a) BK-SDM, Full (b) BK-SDM, LoRA

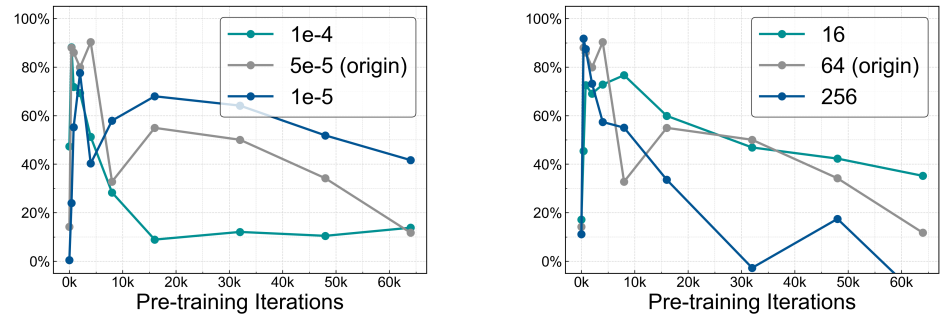
(c) SD-V2, Full (d) SD-V2, LoRA

Figure 11: Statistical significance of fine-tuning speed (AR) across maturity stages. Under full fine-tuning, *Youth* significantly outperforms *Expert* while *Infant–Youth* is not significant; under LoRA, *Expert* is significantly the highest. Error bars denote 90% CIs.

difference is not statistically significant, as their 90% confidence intervals overlap substantially (e.g., Flower: *Infant* 40.26% vs. *Youth* 59.02%; Unsafe: 43.03% vs. 55.66%), indicating that the *Youth* advantage is mainly reflected in the significant decline after the *Youth* stage rather than a significant gain from *Infant* to *Youth*. Under LoRA, *Expert* models achieve the highest AR and are significantly superior to the less mature models: their 90% confidence intervals again do not overlap with those of *Youth* (BK-SDM: Flower 93.76% vs. 63.14%; Cat 73.20% vs. 14.29%; Unsafe 67.97% vs. 31.95%; SD-V2: Flower 89.86% vs. 25.69%; Cat 81.19% vs. 9.86%; Unsafe 59.77% vs. 13.76%). These results confirm that both the full fine-tuning non-monotonic trend and the LoRA monotonic trend are statistically robust.

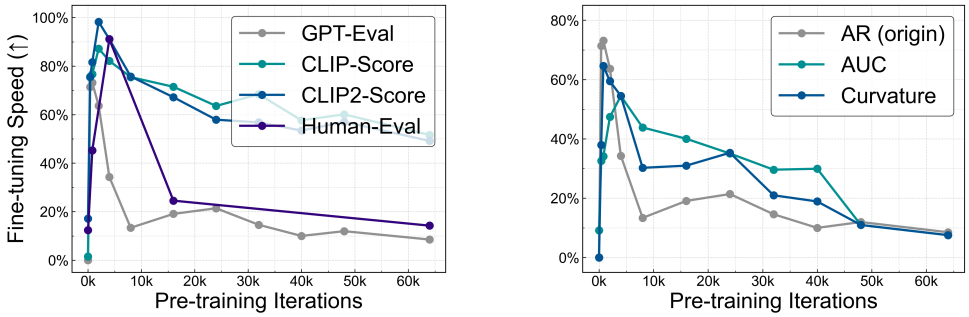
Hyperparameter. To assess the influence of training settings, we vary the learning rate and batch size. As illustrated in Figure 12, higher learning rates and larger batch sizes accelerate the performance decline from Youth to Expert stages, and lower learning rates with smaller batch sizes slow this decline. The fundamental non-monotonic (full fine-tuning) and monotonic (LoRA) patterns remain intact across all settings, confirming robustness to hyperparameter selection.

Evaluation Metrics. As shown in Figure 13a and Figure 13b, we compare multiple image-quality metrics (GPT-Eval, CLIP-Score, CLIP2-Score, Human-Eval) and learning-speed metrics (Area Ratio, AUC, Curvature). The detailed implementations are provided in Section A.3. Although numerical values vary slightly, the fundamental pattern remains consistent that a non-monotonic trend under full fine-tuning versus a monotonic positive correlation under LoRA. Minor variations only affect the exact peak pre-training iteration (0.8k–2k).



(a) Learning Rate (b) Batch Size

Figure 12: The full fine-tuning speed calculated from different hyperparameter setting. The overall trend remains similar.



(a) Image Quality Metric (b) Fine-tuning Speed Metric

Figure 13: The full fine-tuning speed calculated from different evaluation metrics. The overall trend remains similar.

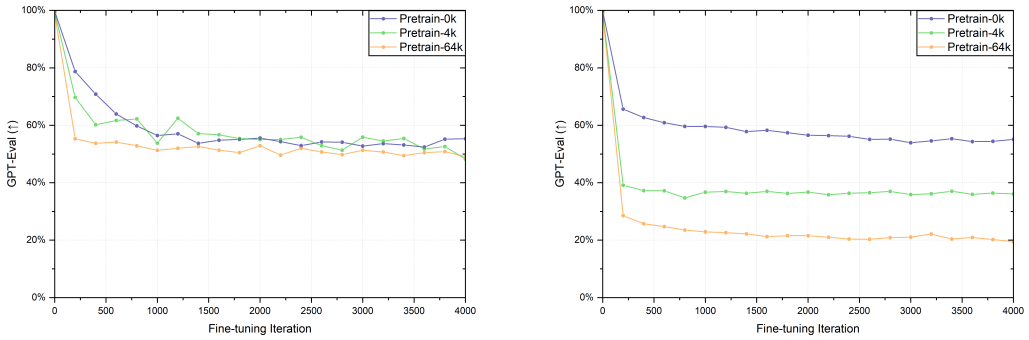
C RQ2: Explanatory Experiment

Figure 14 and Figure 15 show the CLIP2-Score between images generated with prompt "sks113q" and "flower" before and during fine-tuning. In full fine-tuning, the "sks113q" images differ noticeably from those generated with the "flower" prompt, which is reflected in the CLIP2-Score (about 40% in Figure 15). However, the similarity between images generated before and after fine-tuning remains at a high level (about 60% in Figure 14). In contrast, LoRA fine-tuning results in "sks113q" images that closely resemble "flower" images after fine-tuning, with a relatively high CLIP2-Score at approximately 70% in Figure 15.

D RQ3: Cost Analysis

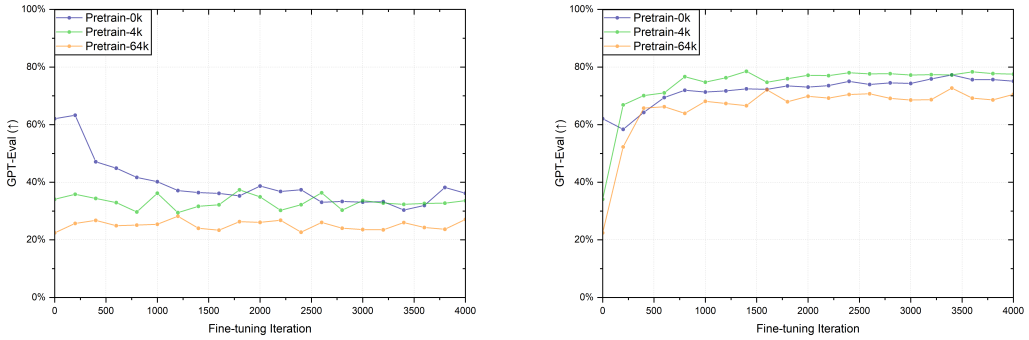
As illustrated in Table 6, for full fine-tuning, selecting a *Youth* checkpoint can reach 95% of best performance within 545 iterations. In contrast, the highest performance is achieved most efficiently by mature *Expert* checkpoints (32k-64k steps), which reach the 95% threshold within 0.24 hours.

Compared with direct downstream training from scratch (0k pre-training), adding a short pre-training phase (0.4k iterations) before fine-tuning significantly reduces the total training time. As shown in Table 7, performing only 400 pre-training iterations before fine-tuning achieves 95% performance in approximately 8.19 hours, showing a substantial improvement over training directly from scratch (25.20+ hours). This corresponds to a cost reduction of more than 67%. Although longer pre-training (e.g., 2k iterations) further accelerates the fine-tuning stage itself, the additional pre-training cost offsets these gains, ultimately resulting in a slightly longer overall training time.



(a) Full (b) LoRA

Figure 14: CLIP2-Score between "sks113q" outputs before and after fine-tuning.



(a) Full (b) LoRA

Figure 15: CLIP2-Score between "sks113q" outputs and "flower" outputs during fine-tuning.

Table 6: The training time (hour) and expense (dollar) of fine-tuning BK-SDM. The least cost is highlighted in bold. In full fine-tuning, using *Youth* checkpoints (8k) can minimize the fine-tuning cost. In contrast, under LoRA fine-tuning, highest performance is achieved most efficiently by *Expert* checkpoints (32k-64k steps).

Method	Threshold	Metric	Pre-training Checkpoints									
			0k	0.4k	0.8k	2k	4k	8k	16k	32k	48k	64k
Full	90%	Avg. FT Step	-	-	-	3778	3329	**545**	3392	2149	2455	2566
		Time (h)	-	-	-	6.75	5.95	**0.98**	6.06	3.84	4.39	4.59
		Expense ($)	-	-	-	2.70	2.38	**0.39**	2.43	1.54	1.76	1.83
	95%	Avg. FT Step	-	-	-	-	-	**545**	3778	-	3273	-
		Time (h)	-	-	-	-	-	**0.98**	6.75	-	5.85	-
		Expense ($)	-	-	-	-	-	**0.39**	2.70	-	2.34	-
LoRA	90%	Avg. FT Step	-	-	-	-	-	-	332	273	**258**	308
		Time (h)	-	-	-	-	-	-	0.25	0.21	**0.19**	0.23
		Expense ($)	-	-	-	-	-	-	0.10	0.08	**0.08**	0.09
	95%	Avg. FT Step	-	-	-	-	-	-	425	**320**	351	352
		Time (h)	-	-	-	-	-	-	0.32	**0.24**	0.26	0.26
		Expense ($)	-	-	-	-	-	-	0.13	**0.10**	0.11	0.11

Table 7: Training time (hours) for SD-V2 fine-tuning from Scratch. Compared with direct downstream training from scratch (0k pre-training), adding a short pre-training phase (0.4k iterations) before fine-tuning significantly reduces the total training time..

Pre. Step	Pre. Time	75%		95%	
		FT	All	FT	All
0k	0.00	17.91	17.91	25.20+	25.20+
0.4k	1.10	5.53	**6.63**	7.09	**8.19**
0.8k	2.20	5.04	7.24	6.87	9.07
2k	5.51	**2.67**	8.18	**3.44**	8.95