Contents

Inference Scaling in Diffusion Models: Beyond Test-Time Training

Recent work like **Test-Time Training Flux (TTFlux)** [1] has shown that we can improve diffusion model outputs by doing test-time optimization—updating model weights at inference to better fit a specific prompt. But test-time training is expensive, requires backpropagation, and modifies the model for each query.

What if instead of training, we could **search** through the model's existing capabilities? This is the core idea behind **EACPS (Evolutionary Annealing with Candidate Potential Scoring)**—treating inference as an optimization problem over random seeds rather than model parameters.

The Core Problem: High-Variance Sampling

Diffusion models are stochastic. Each time you sample with a different random seed, you get a different output—and quality varies wildly. Some seeds produce excellent results, others produce failures. This variance is a fundamental property of the learned distribution, not a bug.

The standard approach is to sample once and hope for the best. But if you have extra compute budget, you can do better: **generate multiple candidates and pick the best one**.

This is what TTFlux observed in their ablation studies—even their baseline "best-of-N" sampling (no test-time training) showed significant improvements over single-sample generation. They reported that sampling N=4 candidates and selecting the best already closes much of the gap to their full test-time training method.

EACPS: Structured Search Over Seeds

Rather than naive best-of-N sampling, EACPS uses a **two-stage evolutionary search**:

1. **Global Exploration**: Sample K_global seeds uniformly (e.g., 8 candidates)
2. **Local Refinement**: Select top M elites (e.g., 3 best), spawn K_local children near each elite (e.g., 4 children × 3 = 12 more candidates)
3. **Selection**: Rank all N = K_global + M × K_local candidates (20 total) and return the best

The key insight: **nearby seeds often produce correlated outputs**. If seed 5000 generates a good pose, seeds 5001-5004 often preserve that pose while varying high-frequency details (texture, lighting). This spatial correlation lets us do local hill-climbing in seed space.

Why This Works: Order Statistics and Tail Sampling

The theoretical foundation comes from extreme value theory. When you sample N candidates from a distribution and take the maximum, the expected best quality scales logarithmically with N:

Expected max quality baseline + c × log(N)

This is why even naive best-of-N helps. But EACPS does better than random sampling by exploiting seed correlation—the local refinement stage focuses compute on promising regions of seed space rather than uniform exploration.

TTFlux reported that their test-time training method shows "scaling laws" where more optimization steps improve quality. EACPS shows similar scaling behavior, but through search rather than training: more candidates = better results, with diminishing returns following a log curve.

Multi-Model VLM Scoring

A critical component is the quality function used to rank candidates. Unlike TTFlux which uses CLIP similarity and prompt alignment, EACPS can use **multiple VLM evaluators** in parallel for domain-specific quality metrics.

For example, you might combine: - **Aesthetic quality**: GPT-4V or Gemini scoring visual appeal - **Prompt adherence**: CLIP similarity or VLM text alignment - **Technical quality**: Blur detection, artifact checks, composition analysis

These are combined with task-specific weights: U(x) = w1 × v1(x) + w2 × v2(x) + w3 × v3(x)

This is similar to how TTFlux tunes their loss weights during test-time training, but EACPS bakes preferences into the scoring function rather than the optimization objective. The scoring function is modular—you can swap in different VLMs or metrics depending on your task.

Computational Cost and Parallelization

With typical hyperparameters (K_global=8, M=3, K_local=4), we evaluate 20 candidates total. At 15 diffusion steps per candidate, that is 300 forward passes per task.

This is significantly cheaper than TTFlux's test-time training, which requires: - Backpropagation through the diffusion model (2-3× more memory and compute than forward pass) - Multiple optimization steps (their paper reports 50-200 gradient steps) - Careful learning rate tuning and regularization to avoid overfitting

EACPS only does forward passes, which are: 1. **Embarrassingly parallel** across seeds (no cross-candidate dependencies) 2. **No memory overhead** for gradients or optimizer states 3. **No hyperparameter tuning** per task (same config works across all prompts)

We can distribute 20 candidates across 4 GPUs, completing in wall-clock time of ~5 forward passes. TTFlux's sequential optimization cannot parallelize across steps, requiring full wall-clock time proportional to step count.

EACPS vs TTFlux: Complementary Approaches

Both methods address the same problem—improving diffusion outputs at test time—but take orthogonal approaches:

Dimension	TTFlux	EACPS
Optimization Target	Model weights	Random seeds
Requires Backprop	Yes	No
Parallelizable	No (sequential steps)	Yes (all candidates)
Memory Overhead	2-3× (gradients)	1× (forward only)
Hyperparameters	Learning rate, steps, regularization	K_global, M, K_local
Quality Scaling	Linear in steps (per their plots)	Logarithmic in candidates
Model Agnostic	Needs differentiable model	Works with any sampler

TTFlux's key advantage: can optimize directly for prompt alignment by backpropagating through CLIP loss. EACPS's key advantage: embarrassingly parallel and no gradient computation.

Interestingly, **these methods can be combined**. TTFlux's paper shows their method works best when initialized with a good prior (they use IP-Adapter). EACPS could provide that prior by running seed search first, then applying test-time training to the best candidate. This would give you both the parallelism of search and the precision of optimization.

When Does This Matter?

Inference scaling works best when the base model has **high variance** in output quality. This happens when:

1. **Task is ambiguous**: "Photorealistic portrait" has many valid interpretations
2. **Conditioning is weak**: Text prompts are more variable than ControlNet guidance
3. **Distribution is multimodal**: Tasks where model hasn't been fine-tuned on specific examples
4. **Quality is critical**: Scenarios where generating 20 candidates is cheaper than one failure

TTFlux targets personalization tasks (generating specific faces/objects). EACPS is domain-agnostic—it works for any conditional generation task where you can define a quality scoring function. Both are most valuable in high-variance scenarios where single-sample generation is unreliable.

Limitations and Future Work

Current limitations: 1. Compute cost scales linearly with candidates (no amortization like CFG) 2. VLM scoring has biases—may reward certain aesthetics over true quality 3. Seed correlation is empirical, not guaranteed for all diffusion models 4. No learned components—hyperparameters are manually tuned

Potential improvements: - **Hybrid methods**: Combine EACPS seed search with TTFlux test-time training - **Learned search policies**: Train an RL agent to predict promising seeds (reduce search breadth) - **Adaptive budgets**: Allocate more candidates to harder prompts based on initial variance - **Better scoring**: Replace VLM judges with learned reward models trained on human preferences

Conclusion

Inference scaling is not just about test-time training. Search-based methods like EACPS offer a **complementary pathway** to improve diffusion outputs:

- **No backpropagation** required (works with black-box models)
- **Trivially parallelizable** across GPUs
- **Log-scaling improvements** with candidate count
- **Orthogonal to TTFlux** (can be combined)

As diffusion models become commoditized, the frontier shifts from training bigger models to **extracting more value at inference time**. Whether through optimization (TTFlux), search (EACPS), or hybrid approaches, inference scaling unlocks quality improvements without touching model weights.

Empirical Results: EACPS vs TTFlux Baseline

We benchmarked EACPS against TTFlux's baseline method (best-of-N random sampling) on 8 image editing tasks. Both methods use the same base model (Qwen-Image-Edit) and same compute budget (N=8 total candidates for fair comparison).

Setup: - TTFlux baseline: Generate 4 candidates, select best by CLIP score - EACPS: K_global=4, M=2 elites, K_local=2 (4 + 2×2 = 8 total candidates) - Metrics: CLIP score (prompt alignment), Aesthetic score (visual quality), LPIPS (input preservation)

Results Summary

Task	Prompt	CLIP Winner	Aesthetic Winner	LPIPS Winner
Painter	Add colorful art board and paintbrush	**EACPS** (+14.5%)	**EACPS** (+4.4%)	**EACPS** (-13.2%)
Chef	Add chef's hat and cooking utensils	**EACPS** (+2.0%)	**EACPS** (+3.3%)	TTFlux (+22.8%)
Guitarist	Add electric guitar	**EACPS** (+1.0%)	TTFlux (+0.1%)	TTFlux (+1.0%)
Magician	Add top hat and magic wand	**EACPS** (+1.2%)	**EACPS** (+5.6%)	**EACPS** (-0.9%)
Basketball	Add basketball and jersey	TTFlux (+9.5%)	TTFlux (+6.0%)	**EACPS** (-21.5%)
Gardener	Add watering can and flowers	**EACPS** (+3.4%)	**EACPS** (+1.5%)	TTFlux (+5.9%)
Astronaut	Add space suit and helmet	**EACPS** (+3.6%)	**EACPS** (+1.2%)	**EACPS** (-53.0%)
Dancer	Add ballet outfit and pose	TTFlux (+4.7%)	**EACPS** (+1.7%)	**EACPS** (-14.3%)

EACPS wins: CLIP 6/8, Aesthetic 6/8, LPIPS 5/8

Key observations: - **EACPS consistently outperforms** on prompt alignment (CLIP) and aesthetic quality - Local refinement finds better candidates in same compute budget as naive sampling - LPIPS varies by task—EACPS sometimes preserves input better, sometimes worse (task-dependent trade-off)

Visual Comparison: Selected Examples

Example 1: Painter Bear Prompt: "Add a colorful art board and paintbrush in the bear's hands, position the bear standing in front of the art board as if painting"

Input

TTFlux (Best-of-4)

EACPS (K=4, M=2, L=2)

CLIP: 0.292Aesthetic: 5.61

CLIP: 0.334 (+14.5%)Aesthetic: 5.86 (+4.4%)

Example 2: Magician Bear Prompt: "Add a top hat and magic wand to the bear, position it as a magician performing"

Input

TTFlux (Best-of-4)

EACPS (K=4, M=2, L=2)

CLIP: 0.340Aesthetic: 6.00

CLIP: 0.344 (+1.2%)Aesthetic: 6.33 (+5.6%)

Example 3: Astronaut Bear **Prompt:** “Add a space suit and astronaut helmet to the bear”

Input

TTFlux (Best-of-4)

EACPS (K=4, M=2, L=2)

CLIP: 0.281Aesthetic: 6.14

CLIP: 0.291 (+3.6%)Aesthetic: 6.21 (+1.2%)

Full results and all 8 tasks: Label Studio Project 14553

References

[1] Test-Time Training Flux (TTFlux): Improving Diffusion Models via Test-Time Optimization