TiPAI: Tournament Inpainting for Patch-Level Alignment in Text-to-Image

Anonymous ACL submission

Abstract

We propose **TiPAI–TSPO**, a decoding-time framework that makes text-to-image generation both *faithful* and *policy-compliant* by editing only where it matters. Starting from a base diffuser, each timestep exposes a low-cost audit view; we mine a small set of suspect regions and run a *tournament of local inpaints* ($N = 5$) produced by an auditor–inpaint model. A learned *auditor–scorer*—trained on 100k DETONATE "selected vs. rejected" pairs—assigns a composite score that integrates text–image faithfulness, policy safety, and seam quality. A *control guard* accepts an edit only if it *beats the unedited control by a calibrated margin* and passes per-class safety/faithfulness thresholds, yielding *monotone non-regression* at the patch level.

To reduce retries and latency, we introduce *Tournament Sampling Policy Optimization (TSPO)*: a light policy over generator knobs (e.g., mask dilation, CFG-inside-mask, latent noise jitter, short-inversion depth) trained with listwise credits and diversity/compute regularizers. TSPO learns to propose winners more often with less compute. We couple edits to the diffusion trajectory via timestep-aware inpainting, short DDIM inversion, and latent-space blending to avoid visual scars. A final calibration stage turns raw scores into reliable acceptance decisions with interpretable knobs (margin δ, policy τ_P, faithfulness τ_F) scheduled across timesteps.

TiPAI–TSPO provides a practical, plug-and-play route to *aligned, efficient* text-to-image decoding.

1 Technical Approach (Step by Step)

We align text-to-image (T2I) decoding by performing *local, timestep-aware edits* that are accepted only when they strictly improve faithfulness and policy safety over the unedited control. The system has three training stages: (A) an auditor–scorer A_s trained on DETONATE pairs to produce global/-patch scores and risk maps; (B) an auditor–inpaint editor A_g pretrained to rewrite local regions, then coupled with a best-of-N tournament and a learned sampling policy (TSPO); and (C) a calibration stage that turns raw scores into reliable decisions with interpretable thresholds.

Preliminaries and Notation

- Prompt p, diffusion trajectory $\{x_t\}_{t=0}^{T}$ or latent $\{z_t\}$ (VAE space).
- Base model produces control proposal at step t: $x_{t-1}^{\text{ctrl}} = \Phi_{\text{base}}(x_t, p, t)$ (or z_{t-1}^{ctrl}).
- An *audit view* I_{t-1} is decoded (possibly at a lower resolution) for patch mining and scoring.
- From I_{t-1}, mine K_t regions $\mathcal{R}_t = \{(R_k, m_k)\}$ using fused saliency: CLIP Grad-CAM drift, detector priors (NSFW/weapon/symbol/logo/age), OCR, and feature-difference heatmaps. Each region has a binary (feathered) mask m; crops are $I_{t-1,R}$.

Stage A: Auditor–Scorer A_s (Global + Patch Ranking with Risk Heads)

Goal. Learn scalar scores $S(p, I)$ and $S_R(p, I_R)$ that order human-*selected* images above *rejected* ones, and produce per-class risk maps for policy guards.

Encoders and Heads.

$$S(p, I) = h_\theta(E_t(p), E_i(I)), \qquad S_R(p, I_R) = h_\theta^{\text{patch}}(E_t(p), E_i(I_R)),$$

with a lightweight decoder that predicts per-class heatmaps $r_c \in [0, 1]^{H\times W}$ from shared features.

Training Data. Pairs $\{(p, I_+, I_-)\}$ from DETONATE. For each pair, mine K patches R_k and assemble $(p, I_{\pm,R_k}, m_{R_k})$. Optional reason codes $y_c \in \{0, 1\}$ per policy class.

Losses (no numbering).

$$\mathcal{L}_{\text{pair}} = \frac{1}{N}\sum_{n}\log(1+\exp(-(S_{+}^{(n)} - S_{-}^{(n)}))),$$

$$\mathcal{L}_{\text{patch}} = \frac{1}{N}\sum_{n}\frac{1}{K_n}\sum_{k}\log(1+\exp(-(S_{R_k,+}^{(n)} - S_{R_k,-}^{(n)}))),$$

$$\mathcal{L}_{\text{policy}} = \frac{1}{N}\sum_{n}\sum_{c}\text{BCE}\Big(\sigma(\bar{\ell}_c(I^{(n)})),\, y_c^{(n)}\Big), \qquad \mathcal{L}_{\text{sal}} = \frac{1}{N}\sum_{n}\sum_{c}\Big(1-\text{Dice}(r_c^{(n)}, \tilde{m}^{(n)})\Big),$$

$$\mathcal{L}_A = \lambda_1\mathcal{L}_{\text{pair}} + \lambda_2\mathcal{L}_{\text{patch}} + \lambda_3\mathcal{L}_{\text{policy}} + \lambda_4\mathcal{L}_{\text{sal}}.$$

Step-by-step recipe.

1. Encode (p, I) and (p, I_R) to obtain global and patch features.
2. Predict S, S_R and risk maps r_c; train with $\mathcal{L}_A$ using a curriculum (start with high-margin pairs, then anneal).
3. Validate with Global Pair-AUC, Patch Pair-AUC, per-class ROC-AUC, and ECE for S.

Outputs used later: calibrated-ready S, S_R, risk maps r_c, reason vector.

Stage B: Auditor–Inpaint A_g + Tournament Sampling Policy Optimization (TSPO)

B1. Pretrain A_g as a timestep-aware local editor. Input formation (context + noise inside mask).

$$X \;=\; I_{t-1}\odot(1-m) \;+\; \epsilon\odot m, \qquad \epsilon\sim\mathcal{N}(0,\sigma_t^2), \qquad t\in\text{chosen timestep band.}$$

Target. $Y = I_R^+$ from the DETONATE selected image.

Inpaint prediction. $\hat{I}_R = A_g(X, m, p, t)$ in RGB or latent.

Losses.

$$\mathcal{L}_{\text{inpaint}} = \beta_1\|\hat{I}_R - Y\|_1 + \beta_2(1-\text{CLIP}(p,\hat{I}_R)) + \beta_3\,\text{Risk}(\hat{I}_R) + \beta_4\,\text{LPIPS}_{\partial m}(\hat{I}_R, Y),$$

$$\mathcal{L}_{\text{pref}} = -\log\frac{\exp(\langle f(\hat{I}_R), f(I_R^+)\rangle/\tau)}{\exp(\langle f(\hat{I}_R), f(I_R^+)\rangle/\tau) + \exp(\langle f(\hat{I}_R), f(I_R^-)\rangle/\tau)}, \qquad \mathcal{L}_G^{\text{pre}} = \mathcal{L}_{\text{inpaint}} + \beta_5\mathcal{L}_{\text{pref}}.$$

Short inversion and latent blending (seam-free coupling). If editing in RGB, map $\hat{I}_R$ back to the scheduler state using a short DDIM inversion of d steps, then blend in latent:

$$z_{t-1} \;\leftarrow\; (1-\alpha m)\odot z_{t-1}^{\text{ctrl}} \;+\; \alpha m\odot z_{t-1,R}^{\text{edit}}, \qquad \alpha=\alpha(t)\in[0,1],$$

with feathered m and timestep-aware α.

Pretrain steps.

1. Sample a pair (p, I^+, I^-), mine R, m, choose a timestep t and form X.
2. Predict $\hat{I}_R = A_g(X, m, p, t)$; compute $\mathcal{L}_G^{\text{pre}}$; update A_g.
3. Validate with patch risk reduction Δr, ΔCLIP on crops, boundary LPIPS on a ring.

B2. Define the tournament and guards (best-of-N + control). Candidate generation. For each region, draw N actions a_i from a generator policy $\pi_\theta(a \mid s)$ with state $s = (p, z_{t-1}, I_{t-1}, m, t)$; produce candidates $C_i = A_g(s; a_i)$. Always include the unedited control C_0.

Scoring. Use A_s to produce, for each i,

$$(S_i,\, F_i,\, P_i,\, B_i) \;=\; A_s(p,\, \text{Compose}(I_{t-1}, C_i, R)),$$

where F is faithfulness, P is policy safety (risk complement), B is seam quality (e.g., $B = \exp(-\kappa\,\text{LPIPS}_{\partial m})$).

Guarded margin utility and selection.

$$u_i \;=\; (S_i - S_0 - \delta)_+ \cdot \mathbf{1}[P_i \ge \tau_P(t)] \cdot \mathbf{1}[F_i \ge \tau_F(t)] \cdot B_i.$$

Let $i = \arg\max_i u_i$. Accept C_i iff $u_i > 0$; otherwise keep C_0 (*control guard* ensures non-regression).

B3. Learn the sampling policy: TSPO. Actions (knobs). Mask dilation/feathering, inside-mask CFG, prompt token emphasis, latent noise jitter, dropout/seed, short inversion depth d, light LoRA routing.

Credits. Leave-one-out advantage for each candidate:

$$A_i \;=\; u_i - \max_{j\neq i} u_j, \qquad \text{or soft credits } w_i = \text{softmax}(u_i/\tau) - \frac{1}{N}.$$

Objective (policy gradient, no numbering).

$$\mathcal{L}_{\text{TSPO}} = -\sum_{i=1}^{N} w_i \log\pi_\theta(a_i \mid s) - \beta\, H[\pi_\theta(\cdot \mid s)] + \lambda_c\,\text{Cost}(\{a_i\}) - \lambda_{\text{div}}\sum_{1\le i<j\le N} d(C_i, C_j),$$

where Cost penalizes short inversions/extra decodes; d is a masked feature distance encouraging within-tournament diversity.

TSPO steps.

1. For each audited region, sample N actions, generate $C_{1:N}$, compute $u_{0:N}$ and pick i.
2. Accumulate $(a_{1:N}, w_{1:N}, \text{Cost}, d)$ and update θ by minimizing $\mathcal{L}_{\text{TSPO}}$ (REINFORCE with entropy/compute/diversity regularizers).
3. (Optional) Improve the judge listwise with a Plackett–Luce/ListNet loss over $\hat{S}_i$ against a target distribution concentrated on the winner.

Stage C: Calibration ⇒ Reliable Decisions

Score calibration on held-out tournaments. Fit isotonic or Platt mapping from raw S to probability $\hat{p}$ on held-out data:

$$\hat{p} = \sigma\left(\frac{S-b}{T}\right) \quad \text{or} \quad \hat{p} = \mathrm{Iso}(S),$$

choosing parameters to minimize NLL and ECE.

Choose operating points (interpretable knobs).

- Margin δ: the smallest Δ such that $\Pr[\text{true win} \mid S_{\text{cand}} \geq S_{\text{ctrl}} + \Delta] \geq 0.9$.
- Policy thresholds $\tau_P(t)$: per-class ROC operating points (stricter late).
- Faithfulness $\tau_F(t)$: stricter mid-trajectory; slightly relaxed at the very end to avoid over-sanitization.
- Seam weight κ and artifact budget (max boundary LPIPS).

Deployed decision rule (per patch, per step).

1. Sample N candidates via π_θ; score with calibrated A_s to obtain (S_i, F_i, P_i, B_i).
2. If there exists a candidate with $S_i \geq S_0 + \delta$, $P_i \geq \tau_P(t)$, $F_i \geq \tau_F(t)$, and B_i above the seam threshold, accept the best one; else keep control and optionally re-audit earlier.
3. Log tournaments for continual TSPO updates and periodic recalibration.

Coupling to the Diffusion Scheduler (Seam-Free Edits)

1. Run audits on a subset $\mathcal{T}_{\text{audit}}$ of timesteps (e.g., every third step), coarse scales early and fine scales late.
2. If A_g edits in RGB, use a short deterministic DDIM inversion of d steps to re-land $\hat{I}_R$ into $z^{\text{edit}}_{t-1,R}$ consistent with the scheduler's $(\bar{\alpha}_t)$ coefficients.
3. Blend in latent with feathered masks and timestep-aware $\alpha(t)$; decode only once per step for the audit view; reuse feature pyramids across candidates.

Complexity and Compute–Quality Tradeoff

For each audited step: cost scales as

$$\mathcal{O}(K_t \cdot (N \cdot C_{A_g} + C_{A_s} + d \cdot C_{\text{DDIMInv}})).$$

Batch inpainting and scoring; cache text/image features. TSPO learns to reduce expected inversions and favors low-cost actions unless higher-cost actions increase the guarded utility.

Properties (Intuition Sketches, No Numbers)

- **Monotone non-regression (patchwise).** The control guard accepts only strict improvements over control that satisfy policy/faithfulness thresholds; otherwise it returns the control, so guarded objectives do not decrease locally.
- **TSPO convergence (fixed judge).** With bounded variance and entropy regularization, policy gradients over a finite action space converge to a stationary point of the expected listwise utility; the compute penalty forms a Lagrangian that places the learned policy on a quality–latency Pareto frontier.
- **Calibration reliability.** Isotonic/Platt calibration on held-out tournaments reduces ECE so fixed (δ, τ_P, τ_F) maintain target operating characteristics across prompts/seeds drawn from the same regime.

Algorithm (Annotated, Step by Step)

1. **Control step:** produce z^{ctrl}_{t-1} (or x^{ctrl}_{t-1}) and decode I_{t-1}.
2. **Mine regions:** build $\mathcal{R}_t = \{(R_k, m_k)\}$ using fused saliency signals.
3. **For each region** (R, m):
   (a) **Generate candidates:** sample N actions $a_i \sim \pi_\theta(\cdot \mid s)$; produce $C_{1:N}$ with A_g; add C_0 (control).
   (b) **Score:** compute (S_i, F_i, P_i, B_i) with A_s (batched).
   (c) **Select:** compute u_i with the guarded margin; if a candidate wins, short-invert and latent-blend into z_{t-1}; else keep control.
   (d) **Log:** store $(a_{1:N}, u_{0:N}, \text{cost}, \text{winner})$ for TSPO.
4. **TSPO update (online or mini-batch):** minimize $\mathcal{L}_{\text{TSPO}}$ with entropy/compute/diversity terms; periodically refine the judge listwise and re-calibrate scores on a held-out slice.

References