INFERENCE-TIME CONTROL FOR SVG GENERATION VIA INFORMATION-PROJECTION GUIDED CONSTRAINED DECODING

Anonymous authors
Paper under double-blind review

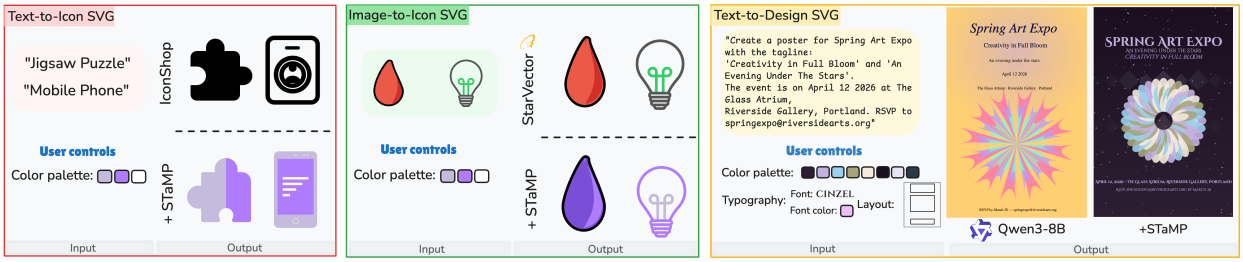

Figure 1: We propose STaMP, an inference time control strategy, which can guide outputs from diverse pre-trained SVG generation LLMs to adhere to color, font and layout controls from a user. Here, we illustrate how STAMP can infuse colors from the palette while generating an icon from text, or while vectorizing an image (first two boxes). In the right-most box, we see how STAMP generates design documents aligned with the user provided controls.

ABSTRACT

Recent autoregressive models can generate SVG from text or images, but they fail to reliably follow user-specified constraints such as colors, layouts and fonts. This limitation highlights that controllability is the missing primitive in autoregressive vector generation. Prompt tinkering and post-hoc edits are brittle, and many practical systems either require retraining for each new constraint or fall back to raster outputs that must be vectorized, underscoring the absence of any autoregressive vector generation method that enables control at inference time. We hypothesize that precise, constraint-driven vector generation is fundamentally a *decoding-time constraint-satisfaction problem*. Formally, we cast this objective as finding the optimal controlled distribution: among all distributions that satisfy the constraints, select the one closest (in KL) to the base model. We show this distribution is the information projection (I-projection) of the base model onto the constrained set. Direct sampling from the I-projection is intractable, but its structure suggests a practical decomposition: a soft reweighting that steers probabilities toward the desired properties and a hard restriction that removes invalid continuations. Building on this insight, we introduce *STaMP (Soft Tilt-and-Mask Policy)*, a model-agnostic, inference-time controller that adds fine-grained control (e.g., color, font, and layout) to any autoregressive SVG model. Evaluated across text-to-SVG and image-to-SVG settings on multiple open models, STaMP delivers inference-time control, consistently improves constraint adherence, and preserves the base model's output quality. Additionally, we introduce, to the best of our knowledge, the first text-to-design SVG model as an extended showcase: paired with STaMP, it produces full compositions as structured, editable SVG while honoring user-defined controls over color, typography, layout, and asset placement, all within a single inference pass.

1 INTRODUCTION

Modern design workflows increasingly rely on vector graphics for their resolution independence, editability, and compact representation, yet generating production-ready SVG (Scalable Vector Graphic) designs from natural language remains fundamentally limited by lack of control Polaczek et al. (2025). While recent autoregressive models can generate impressive vector illustrations from text prompts, they operate as black boxes: once generation begins, designers cannot intervene to enforce brand colors, adjust layouts, or ensure specific typography without regenerating from scratch or manual post-editing. Existing solutions (Thamizharasan et al., 2024; Zhang et al., 2024) either require costly model retraining for each new constraint set, resort to generating raster images that must be vectorized (losing

the benefits of native vector generation), or rely on prompt engineering that offers no guarantees of constraint satisfaction. At the other extreme, rule-based approaches that strictly enforce constraints produce rigid, uncreative outputs that fail to leverage the generative model's learned design knowledge (Dathathri et al., 2019; Yang & Klein, 2021; Krause et al., 2020). This paper, therefore, addresses a precise challenge: given any pretrained autoregressive SVG model, can we develop an inference-time control mechanism that enforces user-specified constraints while preserving the model's creative capabilities, without any retraining?

Classical fixes attack symptoms rather than the cause. Online fixes treat violations as sampling noise or try to nudge the decoder on the fly. Prompt engineering, temperature or seed sweeps, and rejection sampling is compute heavy, collapse diversity, and still offer no guarantees. Ad-hoc logit biasing warps calibration, invites syntax errors, and pushes the model off its learned manifold. Constrained beam search and post-hoc validators shift the burden into search, where non-local constraints cause hypothesis sets to explode and pruning reintroduces brittleness. Grammar-based decoders enforce syntax but cannot coordinate long-range relations and quickly flatten the model's priors into templated outputs. Offline fixes try to pre-bake constraints into the system. Per-constraint finetuning must cover a combinatorial space of rules, so each retrain is slow, drifts with time, and fails on unseen combinations of controls . Raster-first pipelines avoid structure and vectorize later, which yields tangled paths, oversized files, and no semantic link between code tokens and design elements, making enforcement and auditing impossible. Template retrieval guarantees conformity at the cost of homogenization. Reinforcement-learning decoders demand heavy data and delicate reward shaping, and at test time still need safety shields to keep code valid. In short, online methods treat symptoms and offline methods hard-code them, and neither provides a principled way to shape the next-token distribution so that control is built in, not bolted on as an afterthought.

The fundamental limitation is that prior fixes treat control as a pre-processing problem (rewrite the prompt) or a post-processing problem (repair the output), when what is needed is control *during* generation. Autoregressive SVG models are attractive because they emit executable vector code, cover diverse styles under simple conditioning, and fit naturally into design-as-code workflows. Their weakness is structural: once decoding begins there is no mechanism to enforce user-specified controls. Training a new model or adding special heads for every control is not feasible at scale and does not generalize across scenarios. What has been overlooked is that these models expose, at every step, the full next-token distribution–a probability over all continuations–which in principle indicates which continuations keep the requested controls still satisfiable; using that information in real time is the hard part. Building upon this insight, we propose the *first training-free, model-agnostic, inference-time controller* to align the output token distribution to that of the conditions from the user. Our intuition is simple: if we shift probability toward continuations that keep the controls satisfiable and rule out next-token choices that would make them unattainable, decoding stays close to the base model while meeting the controls. We formalize this by projecting the base distribution onto the constraint-satisfying set, and implement that projection during decoding via a soft probability tilt with a deterministic mask of impossible continuations.

Our contributions: (***i***) We reformulate controlled vector generation as a decoding-time constraint-satisfaction problem and give a principled characterization of the optimal controlled distribution as an information projection of the base model. This yields a clean decomposition of control into soft reweighting and hard support restriction. (***ii***) We introduce STaMP (Soft Tilt-and-Mask Policy), a model-agnostic, retraining-free controller that operates on logits to softly tilt next-token probabilities toward constraints while deterministically masking invalid continuations, preserving SVG correctness and long-range structure. (***iii***) We show that STaMP confers practical, fine-grained control across all open autoregressive SVG generators we test covering both text-to-SVG and image-to-SVG consistently improving constraint satisfaction and code cleanliness with minimal overhead. (***iv***) To stress-test controllability, we build the first text-to-design SVG model that outputs full compositions (e.g., posters, business cards) as clean, editable code, and demonstrate that STaMP enforces user-defined controls in a single pass.

Note: We use "control" to mean user-specified constraints on the SVGs. In this paper we instantiate three canonical controls – color palette, typography, and layout – because these dimensions most strongly distinguish vector graphics and can be evaluated directly from SVG structure and rendering semantics (Polaczek et al., 2025). Our framework is control-agnostic and applies to any property with a computable scorer or recognizer. Palette, typography and layout are representative instantiations used to ground experiments, and our methodology can scale to controls beyond these.

2 Related works

Autoregressive SVG generation: Early sequence and VAE-style sketch models showed that vector graphics can be emitted as code (autoregressively) (Ha & Eck, 2017; Cao et al., 2019; Ribeiro et al., 2020; Lopes et al., 2019), and document-level variants expanded compositional scope across shapes and paths (Carlier et al., 2020; Yamaguchi,

2021). Code LLMs improve numeric precision and compositional control (Wu et al., 2023; Xing et al., 2025; Wang et al., 2025), and multimodal systems broaden semantics via visual tokens or draft–refine pipelines (Rodriguez et al., 2023; Yang et al., 2025b; Wu et al., 2025). However, these models run open-loop at inference time, so control remains prompt-level or post-hoc (reranking/filters), not a policy that guarantees target satisfaction during sampling. Vectorization methods solve the inverse problem from rasters with differentiable rendering or structured fitting (Reddy et al., 2021; Hu et al., 2024; Ma et al., 2022), but they are not next-token policies and thus cannot offer enforceable per-step controls. Optimization approaches–gradient-guided curves (Frans et al., 2022; Schaldenbrand et al., 2021; Vinker et al., 2022) and diffusion-to-vector transfers (Jain et al., 2023; Xing et al., 2023; 2024)–achieve high fidelity via iterative objectives, but lack amortized next-token distributions and provide no mechanism to impose exact sequence-level targets while decoding. Geometry-aware latents reduce path tangling by representation design (Thamizharasan et al., 2024; Zhang et al., 2024), yet operate at training/encoding time rather than offering run-time, token-level control. In contrast, we introduce a decoding-time controller for any autoregressive SVG model that enforces per-step constraints and calibrates sequence-level targets through explicit, information-theoretic reweighting, moving from prompt-only steering to guaranteed control at generation time.

Decoding-time control: Attribute-guided decoding steers next-token distributions using auxiliary signals—gradient injections or prefix/future discriminators (Dathathri et al., 2019; Yang & Klein, 2021), Bayes mixing with a class LM (Krause et al., 2020) or expert/anti-expert logit composition (Liu et al., 2021). These are effective for coarse attributes but remain *soft*: they hinge on calibrated scorers, do not enforce instance-specific structural targets (e.g., nested tag obligations or element counts), and can trade fluency for control when pushed. Energy-based constrained decoding frames control as minimizing a sequence energy with Lagrange multipliers or Langevin dynamics (Kumar et al., 2021; 2022b; Qin et al., 2022; Liu et al., 2023). This enables stronger steering but relies on differentiable surrogates, iterative inner loops, and does not expose a streaming left-to-right policy with structural guarantees. MCMC (Gonzalez et al., 2025) and edit-based controllers satisfy black-box scorers via token/block-level Metropolis–Hastings or progressive rewrites ((Mireshghallah et al., 2022; Forristal et al., 2023; Yu et al., 2023; Hallinan et al., 2022), but require many proposals, struggle with rare events, and lack amortized next-token distributions. Latent composition and posterior-regularization methods (Dekoninck et al., 2023; Zhong et al., 2023; Meng et al., 2022) decompose sequence-level oracles into token-level guidance and improve balance/coverage, yet they presume reliable oracles and still do not provide per-step viability under nested syntax. In short, existing controllers either steer attributes softly or achieve control via slow, iterative sampling; none directly couples a streaming next-token policy with explicit structural enforcement and calibrated sequence-level targets, which is precisely the system needed for controlled SVG code generation.

Constrained decoding: Grammar- and automata-based decoding constrains token streams to a formal language using persistent parse stacks or closed-form automata (Dong et al., 2024; Koo et al., 2024), with token–grammar alignment and speculative execution improving latency and coverage (Beurer-Kellner et al., 2024; Park et al., 2025), and prefix pruning enforcing structure on the fly (Sun et al., 2025; Scholak et al., 2021). These mechanisms are strong for syntax in code, but they (i) often certify only prefix acceptance rather than *viability* to an accepting terminal under length/EOS budgets, (ii) depend on tight grammar–subword alignment that can be brittle when tags/attributes straddle tokens, and (iii) cannot natively encode cross-field numeric/geometric relations (e.g., coordinate consistency) that lie outside regular and many context-free classes. Controllers based on scopes, types, and static analysis (Ugare et al., 2024b; Dong et al., 2022; Mündler et al., 2025; Poesia et al., 2022; Agrawal et al., 2023) raise semantic correctness but still require domain-specific analyzers and do not provide per-step calibration of numeric attributes or global element counts. Lexically constrained decoding (Hokamp & Liu, 2017; Post & Vilar, 2018; Hu et al., 2019; Bogoychev & Chen, 2023; Anderson et al., 2016; Lu et al., 2020; Lin et al., 2019) guarantees inclusion of required tokens, yet inclusion alone neither ensures balanced structures nor prevents dead-ends; aggressive term forcing can also shrink the viable set and degrade search. Sampling and refinement frameworks—token/block MCMC and posterior methods (Su et al., 2018; Lipkin et al., 2025; Lew et al., 2023; Ye et al., 2025; Park et al., 2024; Zhang et al., 2023) and gradient-based or backtracking refinements (Qin et al., 2022; Kumar et al., 2022a; Agarwal et al., 2025a; Ugare et al., 2024a; Geng et al., 2024; Banerjee et al.; Le et al.; Li et al., 2024; Choi et al., 2023; Hemmer et al., 2023)–enforce constraints post hoc but rely on iterative proposals or inner loops and do not expose a single-pass, amortized next-token policy with structural guarantees under nested, numeric SVG requirements. Our approach leverages grammar-aligned masking for structure but couples it with explicit viability and a probabilistic steering term, yielding a streaming policy suited to SVG's nested and numeric constraints.

3 I-Projection Guided Constrained Decoding

Problem statement: Let $\mathcal{V}$ be a finite token vocabulary with a designated end-of-sequence token $\langle \text{eos} \rangle$. An autoregressive SVG generator defines a distribution over variable-length sequences $x = (x_1, \ldots, x_{|x|}) \in \mathcal{V}^*$ and assigns joint

probability by the chain rule,

$$P_\theta(x) \;=\; \prod_{t=1}^{|x|} P_\theta(x_t \mid x_{<t}), \qquad x_{<t} = (x_1, \ldots, x_{t-1}). \tag{1}$$

We use the standard transformer-style decoding interface with cached history

$$\mathcal{H}_t : (o_{t+1},\, \mathcal{H}_{t+1}) = \mathrm{LM}_\theta(x_t, \mathcal{H}_t) \tag{2}$$

with $x_{t+1} \sim \mathrm{Softmax}(W o_{t+1})$. Let Σ denote the character alphabet. A deterministic decoder $\mathrm{dec} : \mathcal{V}^* \to \Sigma^*$ maps a token sequence x to its SVG source string, $\mathrm{SVG}(x) \coloneqq \mathrm{dec}(x)$, which the renderer consumes.

Our goal is to *control* this generic autoregressive generator at inference time. This is not straightforward: SVG obeys strict well-formedness, many constraints are long-range and only decidable after complete rendering, and tokenization may split tag lexemes across multiple tokens. Naïve next-token rules can admit dead-ends or silently violate global structure, while purely global reweighting lacks a left-to-right factorization.

We therefore separate requirements into hard and soft constraints. Hard constraints encode syntactic/structural validity via a recognizer $\mathcal{R}$ (e.g., a stack-based XML checker). The feasible token set then is $\mathcal{C} \;=\; \{\, x \in \mathcal{V}^* : \mathcal{R} \text{ accepts } \mathrm{SVG}(x) \,\}$. Soft constraints capture style targets through a utility $f : \mathcal{V}^* \to \mathbb{R}^m$ with desired moment $c \in \mathbb{R}^m$ (enforced elementwise). We assume feasibility: $P_\theta(\mathcal{C}) > 0$ and $c \in \{E_Q[f] : Q \ll P_\theta,\ Q(\mathcal{C}) = 1\}$.

We want the controlled generator to (i) place all mass on valid SVGs, (ii) hit the soft target in expectation, and (iii) change the base model as little as possible so fluency and prior knowledge are preserved. Among all distributions satisfying (i)-(ii), the forward KL, $\mathrm{KL}(Q\|P_\theta)$, implements the *minimum-information change* to P_θ; it yields a unique Bregman (information) projection and guarantees absolute continuity with respect to P_θ. It is also behaviorally appropriate: forward KL is mode-covering (preserves diversity) and, in the special case with only hard constraints, reduces exactly to conditioning on validity. These desiderata lead to the information-projection problem

$$\min_{Q \in \Delta(\mathcal{V}^*)} \mathrm{KL}(Q \,\|\, P_\theta) \;\; \text{s.t. hard}: \; Q(\mathcal{C}) = 1, \;\; \text{soft}: \; E_Q[f(x)] = c. \tag{3}$$

Concretely, we seek a model-agnostic token-level *control policy* that enforces structural well-formedness *at inference-time*, and biases sampling toward the desired semantics without retraining the backbone.

3.1 Information-Projection Solution

How should we solve equation 3? A direct search over distributions is infeasible, but equation 3 has a closed-form optimizer once we introduce Lagrange multipliers for the soft moments and normalization and restrict attention to sequences in $\mathcal{C}$. Writing the Lagrangian:

$$\mathcal{L}(Q, \eta, \mu) = \sum_{x \in \mathcal{C}} Q(x) \ln \frac{Q(x)}{P_\theta(x)} \;-\; \eta^\top \Big(E_Q[f] - c \Big) \;+\; \mu \left(\sum_{x \in \mathcal{C}} Q(x) - 1 \right), \tag{4}$$

with $\eta \in \mathbb{R}^m$ and $\mu \in \mathbb{R}$. The stationarity condition $\partial \mathcal{L} / \partial Q(x) = 0$ yields, for every $x \in \mathcal{C}$,

$$\ln Q^*(x) - \ln P_\theta(x) + 1 - \eta^\top f(x) + \mu = 0 \quad \Longrightarrow \quad Q^*(x) = \tfrac{1}{Z(\eta)}\, P_\theta(x)\, \exp\big(\eta^\top f(x)\big), \text{ and } Q^*(x) = 0 \text{ for } x \notin \mathcal{C},$$

where the partition function

$$Z(\eta) \;=\; \sum_{x \in \mathcal{C}} P_\theta(x) \exp\big(\eta^\top f(x)\big) \tag{5}$$

ensures normalization. Thus the I-projection is an exponential tilting of the base model, restricted to valid sequences.

Moment matching and dual problem. Define $\psi(\eta) = \log Z(\eta)$. Standard properties of the log-partition imply

$$\nabla \psi(\eta) \;=\; E_{Q_\eta}[f(x)] \quad \text{and} \quad \nabla^2 \psi(\eta) \;=\; \mathrm{Cov}_{Q_\eta}\big(f(x)\big) \;\succeq\; 0, \tag{6}$$

so the multiplier η^* is determined by the moment-matching condition $E_{Q_{\eta^*}}[f] = c$. Equivalently, η^* maximizes the concave dual:

$$\max_{\eta \in \mathbb{R}^m} \; \underbrace{\eta^\top c - \log Z(\eta)}_{\text{dual objective}}, \tag{7}$$

whose gradient is $c - E_{Q_\eta}[f]$ and whose Hessian is $-\mathrm{Cov}_{Q_\eta}(f) \preceq 0$. Under the feasibility assumption in the problem statement, the primal optimum Q^* exists and is unique (KL is strictly convex in Q on the feasible set). See Appendix A.3 for existence/uniqueness and multiplier calibration details.

Conditionals of the optimal distribution. Although equation 5 gives the joint form of Q^*, generation is left-to-right. For any prefix $x_{<t}$ with nonzero Q^*-mass, the optimal next-token conditional factors as

$$Q^*(x_t \mid x_{<t}) \;=\; P_\theta(x_t \mid x_{<t}) \cdot \frac{G_\eta(x_{<t}x_t)}{G_\eta(x_{<t})}, \qquad G_\eta(x_{<t}) \;=\; E_{x_{t:}\sim P_\theta}\Big[\exp\big(\eta^\top f(x)\big)\,\mathbf{1}_{\{x\in\mathcal{C}\}} \;\Big|\; x_{<t}\Big], \tag{8}$$

where $G_\eta(\cdot)$ is the completion partition, a conditional log-moment-generating function over all valid continuations. Two immediate sanity checks follow from equation 8: (i) with only the hard constraint ($\eta = 0$), Q^* reduces to P_θ conditioned on $\mathcal{C}$; (ii) with only the soft constraint and $\mathcal{C} = \mathcal{V}^*$, Q^* reduces to a global exponential tilt of P_θ.

Why this matters for decoding? Equations equation 5-equation 8 characterize the *ideal* controlled generator defined by equation 3. In principle, sampling from Q^* would exactly satisfy the hard validity requirement and meet the soft moment target while minimally deviating from P_θ.

Intractability of the direct solution. Direct realization of Q^* is computationally prohibitive for autoregressive SVG generation:(i) *Global normalization:* Computing $Z(\eta)$ (and hence solving the moment equation) requires summing $P_\theta(x)\exp(\eta^\top f(x))$ over all valid sequences $x \in \mathcal{C}$, an exponentially large space. (ii) *Nonlocal conditionals:* The factor $G_\eta(x_{<t})$ in equation 8 aggregates all valid completions of a prefix under P_θ, coupling the next-token choice to the entire suffix. Exact evaluation (or even tight approximation) is generally intractable; naive rejection or importance sampling degenerates for rare-event constraints. These obstacles preclude computing or sampling from Q^* exactly during left-to-right decoding, and they motivate the approximate, online controller introduced in the next section.

4 STAMP: SOFT TILT–AND–MASK POLICY

The I-projection in equation 3 specifies the ideal constrained generator, but equation 5–equation 8 make clear that exact normalization and next-token conditionals are nonlocal and intractable during left-to-right decoding. STaMP is a decoding-time controller that emulates the two ingredients implicit in equation 8: an exponential reweighting toward the target (soft) and a restriction of support to the valid set (hard), realized online as a Soft Tilt followed by a Mask.

4.1 SOFT TILT: LOCAL APPROXIMATION TO THE OPTIMAL CONDITIONAL

The optimal conditional in equation 8 factors $Q^*(x_t \mid x_{<t})$ into the base next-token model and a multiplicative term

$$\frac{G_\eta(x_{<t}x_t)}{G_\eta(x_{<t})} \;=\; \exp\Big(\log G_\eta(x_{<t}x_t) - \log G_\eta(x_{<t})\Big),$$

where $G_\eta(\cdot)$ is the completion partition over valid continuations. We approximate the corresponding ideal log-bias $b^*(x_{<t}, x_t) := \log \frac{Q^*(x_t|x_{<t})}{P_\theta(x_t|x_{<t})} = \log G_\eta(x_{<t}x_t) - \log G_\eta(x_{<t})$ with a learned function $b_\phi(x_{<t}, x_t)$ produced by an adapter that observes the model's history state $\mathcal{H}_t$ (cf. equation 2). The Soft Tilt forms a biased one-step distribution

$$P_{\text{tilt}}(x_t \mid x_{<t}) \;\propto\; P_\theta(x_t \mid x_{<t})\,\exp\big(b_\phi(x_{<t}, x_t)\big), \tag{9}$$

which is an autoregressive, locally factorized surrogate for the soft reweighting induced by the I-projection. For vector-valued utilities $f \in \mathbb{R}^m$, b_ϕ can be parameterized to approximate $\eta^\top$ times a predicted incremental contribution to f; the scalar η (or vector, elementwise) is the same Lagrange multiplier from equation 4–equation 7 and is calibrated to target $E[f] = c$.

4.2 MASK: VIABILITY-PRESERVING HARD ENFORCEMENT

To realize $Q(\mathcal{C}) = 1$ (hard in equation 3) during decoding, we maintain a recognizer $\mathcal{R}$ for well-formed SVG (stack-based tag matching plus finite-state attribute/lexical checks). At time t, given the current prefix $x_{<t}$, let $\mathcal{V}_t \subseteq \mathcal{V}$ be the set of tokens whose emission keeps *some* valid completion reachable (viability). The Mask operation zeroes the probability of all other tokens:

$$P_{\text{tilt+mask}}(x_t \mid x_{<t}) \;\propto\; P_{\text{tilt}}(x_t \mid x_{<t})\,\mathbf{1}_{\{x_t\in\mathcal{V}_t\}}, \qquad \mathcal{V}_t \;=\; \{v \in \mathcal{V} :\; \mathcal{R} \text{ stays viable on } x_{<t}v\}. \tag{10}$$

This enforces hard validity at every step (soundness); see Appendix A.1. If equation 10 uses exact viability (accounting for EOS/length budgets), sampling with Mask alone is equivalent to conditioning P_θ on $\mathcal{C}$ when $\eta = 0$. A formal proof of this equivalence appears in Appendix A.2. Combined with equation 9, it implements the two structural factors of equation 8 in a left-to-right form. Under oracle tilt and exact viability, the policy equals Q^*; see Appendix A.6.

4.3 Engineering Choices and Implementation Recipe

STaMP has three moving parts: a recognizer $\mathcal{R}$ that answers viability queries, an adapter producing the Soft Tilt, and a small loop that composes them during decoding. The recognizer is implemented as a deterministic pushdown automaton for balanced, properly nested tags, augmented with a finite-state layer for attribute syntax and forbidden/required substrings; it exposes a query that returns $\mathcal{V}_t$ for a given prefix and budget. The adapter reads the history state $\mathcal{H}_t$ and emits a residual logit $b_\phi(x_{<t}, \cdot)$; magnitude constraints (e.g., clipping or temperature scaling on b_ϕ) prevent softmax saturation and preserve calibration.

Training follows the logic of equation 3 but replaces the intractable sequence-level divergence with a token-level surrogate. Let h_t denote the decoder state (including $\mathcal{H}_t$ and prefix features) distributed according to the rollout policy $\tilde{Q}_\phi$ induced by equation 9+ equation 10, and let $\pi_\phi(\cdot \mid h_t)$ be the corresponding next-token distribution after Soft Tilt and Mask. We optimize:

$$\max_{\phi} \; \eta^\top E_{x\sim\tilde{Q}_\phi}[\, f(x)\,] \;-\; \sum_t E_{h_t\sim\tilde{Q}_\phi}\Big[\,\mathrm{KL}\big(\pi_\phi(\cdot \mid h_t) \,\|\, P_\theta(\cdot \mid h_t)\big)\Big], \qquad \pi_\phi \text{ induced by equation 9 + equation 10.}$$

By Appendix A.5, this per-step KL equals the sequence-level $\mathrm{KL}(\tilde{Q}_\phi \,\|\, P_\theta)$. The first term drives the soft moment toward c and the per-step KL enforces proximity to the base next-token policy. Gradients are estimated with policy-gradient (score-function) methods using per-step baselines; alternatively, one may also distill from reward-weighted trajectories by minimizing cross-entropy between $P_{\text{tilt}}(\cdot \mid x_{<t})$ and empirical, reward-weighted next-token counts.

Calibration of η proceeds by a outer loop: for a candidate η, decode short batches with the current controller, estimate $E_{\tilde{Q}_\phi}[f]$, and update $\eta \leftarrow \eta + \alpha\,(c - E_{\tilde{Q}_\phi}[f])$ (componentwise for $m > 1$). Under the monotonicity implied by equation 6, this one-dimensional search converges to the desired moment. At inference time, the controller can run at every step, but SVG's structure makes selective activation more efficient. We gate STaMP by the recognizer's lexical state and engage it only at semantic decision points, e.g., on entering or closing a tag, emitting path data etc., while otherwise sampling directly from P_θ. When engaged, we compute base logits, add the Soft Tilt residual, query $\mathcal{V}_t$ from $\mathcal{R}$, apply the Mask, and sample; this reduces average cost to $O(\rho\,|\mathcal{V}|)$ arithmetic per token (with $\rho \in [0, 1]$ the fraction of controlled steps) plus a constant-factor automaton cost. With feasible constraints, the viable set remains nonempty and the loop proceeds without backtracking; optional heuristics (temperature, top-k, nucleus) may be layered on but constitute further deviations from the implied $\tilde{Q}_\phi$. Theoretical guarantees and proofs are shown in the Apendix.

5 Results

We structure results around three research questions (RQ) that evaluate *controllability* rather than isolated metrics. RQ1 Model-agnostic control: Can STaMP enforce constraints on *any* autoregressive SVG model without requiring retraining? RQ2 End-to-End designs: In a more challenging setting, can STaMP handle design specifications by supporting the first text-to-SVG design models, thus serving as an end-to-end control stress test? RQ3 Comparative control: How does STaMP compare against strong alternatives in the quality–satisfaction–efficiency trade-off?

5.1 Model-Agnostic Control

Experiment setting: We pair STaMP with publicly available text-to-SVG and image-to-SVG base models: OmniSVG (Yang et al., 2025b), LLM4SVG (Xing et al., 2025) (with Qwen 2.5 and Gemma 3 backbones), IconShop (Wu et al., 2023), and StarVector Rodriguez et al. (2023). For StarVector, we evaluate only the image-to-SVG (the text-to-SVG model is not publicly released). In addition, we train Qwen 3 (Yang et al., 2025a) and GPT-OSS (Agarwal et al., 2025b) on a proprietary design SVG corpus and include them as backbones. All evaluations use OmniSVG's MMSVG-Icon subset from MMSVG-Bench (Yang et al., 2025b), restricting to icon-level tasks to enable fair comparisons across backbones. This avoids conflating results with model size disparities (the publicly released OmniSVG checkpoint is smaller than the strongest models reported), which would otherwise dominate outcomes in broader settings.

Evaluation protocol: Our goal is not to rank base models, but to test whether pairing each backbone with STaMP yields reliable control *without* deteriorating the backbone's strengths. We report control metrics–CIEDE 2000 (Schanda, 2007) (palette control), overlap % between specified and realized regions (layout control), and font matching for typography–and general metrics that should remain stable under control: quality retention (LPIPS Zhang et al. (2018), token complexity (number of SVG tokens), generation time, and CLIP score. For RQ1, we focus on two controls and their combination; for each backbone we compare its native decoding against the same model wrapped with STaMP under identical prompts and budgets.

Table 1: Impact of STaMP on controllability of text-to-SVG models: baselines with STaMP vs without, evaluated on C (color), L (layout), and C+L. Arrows indicate the favorable direction per metric.

Model	Variant	C: Color Constraint				L: Layout Constraint				C+L Constraints				
		CIEDE 2000 ↓	Complexity (# Tokens) ↓	Generation Time (s) ↓	CLIP Score ↑	Overlap % ↑	Complexity (# Tokens) ↓	Generation Time (s) ↓	CLIP Score ↑	CIEDE 2000 ↓	Overlap % ↑	Complexity (# Tokens) ↓	Generation Time (s) ↓	CLIP Score ↑
OmniSVG [3B]	Base	6.41	3.32k	15	0.3012	7.18	3.35k	16	0.3020	6.73	8.62	3.40k	16	0.3019
	+STaMP	0.05	3.38k	21	0.3015	84.95	3.99k	21	0.3000	0.02	88.54	3.82k	22	0.3002
LLM4SVG Qwen2.5 [7B]	Base	8.63	2.10k	25	0.2498	3.17	2.13k	26	0.2489	9.31	4.19	2.15k	26	0.2491
	+STaMP	0.07	2.35k	31	0.2483	88.82	2.89k	33	0.2488	0.04	89.76	2.94k	35	0.2487
LLM4SVG Gemma 3 [4B]	Base	13.91	1.91k	16	0.2109	4.24	1.94k	16	0.2109	13.38	4.75	1.94k	17	0.2110
	+STaMP	0.06	2.13k	24	0.2100	84.16	2.50k	27	0.2101	0.01	85.69	2.53k	26	0.2109
IconShop	Base	32.04	3.38k	7	0.2079	3.12	3.38k	7	0.2077	38.05	3.08	3.39k	7	0.2076
	+STaMP	0.13	3.51k	19	0.2094	86.47	4.17k	20	0.2103	0.08	85.32	4.21k	21	0.2102
Qwen3 [8B]	Base	5.18	5.18k	51	0.3103	31.25	5.92k	63	0.3100	5.64	30.61	6.01k	65	0.3102
	+STaMP	0.03	5.20k	59	0.3096	96.17	6.42k	78	0.3099	0.05	94.13	6.51k	77	0.3100
GPT-OSS [20B]	Base	3.79	5.31k	68	0.3321	48.62	6.01k	77	0.3322	3.85	46.03	6.18k	78	0.3322
	+STaMP	0.02	5.31k	87	0.3323	98.14	6.58k	91	0.3323	0.07	97.26	6.82k	96	0.3324

5.1.1 RQ1 Key Results:

Key result #1: STaMP enables model- and modality-agnostic inference-time control. Tables 1 and 2 report the quantitative results, and Figure 1 visualizes the same effect: STaMP enforces palette and layout across both text-to-SVG and image-to-SVG models, independent of the backbone. Palette is decided by a small, discrete set of attribute SVG tokens. The soft tilt concentrates probability on those tokens at the moments they matter, and calibration pins the palette error near-zero without touching geometry. Layout is decided by coordinate/path tokens spread across the sequence. The viability mask prunes choices that would make the required placement unreachable under the remaining length/stack budget, so geometry snaps toward the requested arrangement rather than wandering. These two controls largely act on disjoint token subsets, so applying them together is close to commutative in practice. STaMP works because it sits at the same next-token/logit interface in both text-to-SVG and image-to-SVG. In each case, once the backbone produces the per-step distribution, STaMP applies a soft tilt to favor palette/layout-consistent tokens and a mask to remove choices that break feasibility; nothing upstream (encoders, prompts, or training) is touched. Operating at this decoding neck makes the mechanism inherently model- and modality-agnostic, which is why the same controller behaves consistently across all backbones.

Table 2: Image-to-SVG comparison under controllability constraints. Baselines with STaMP vs without.

Model	Variant	C		L		C+L
		CIEDE 2000 ↓	Overlap % ↑	Quality Retention ↑	Complexity (# Tokens) ↓	Generation Time (s) ↓
StarVector [8B]	Base	11.6	9.3	0.11	3.72k	49
	+STaMP	0.1	84.1	0.13	4.26k	58
OmniSVG [3B]	Base	10.9	8.2	0.15	3.92k	18
	+STaMP	0.08	78.9	0.15	4.52k	26
StarVector [8B]	Base	11.6	9.3	0.13	3.72k	49
	+STaMP	0.1	84.1	0.16	4.26k	58
OmniSVG [3B]	Base	10.9	8.2	0.13	3.92k	18
	+STaMP	0.08	78.9	0.15	4.52k	26
StarVector [8B]	Base	11.6	9.3	0.14	3.72k	49
	+STaMP	0.1	84.1	0.16	4.26k	58
OmniSVG [3B]	Base	10.9	8.2	0.15	3.92k	18
	+STaMP	0.08	78.9	0.16	4.52k	26

Key result #2: STaMP unlocks *zero-shot* color, typography, and layout control on a monochrome-only backbone. IconShop (Wu et al., 2023) is trained to generate monochrome icons and by design, does not exercise color or typography controls. Paired with STaMP and stressed with *all three* controls at once, it produces SVGs that adopt the specified palette, insert the requested text with the correct font, and place elements in the required arrangement, as shown in Fig. 2. The intuition is simple: STaMP sits at the decoding neck and acts on the backbone's own next-token probabilities, even if the backbone rarely uses color or text tokens, they remain in the vocabulary with nonzero mass, the Soft Tilt lifts them precisely when needed, and the Mask keeps future completion feasible. This converts "unused but available" (long tail) capabilities into reliable, decode-time control - without retraining the model or changing its encoder–and the same zero-shot effect appears whenever the relevant tokens and grammar are present.

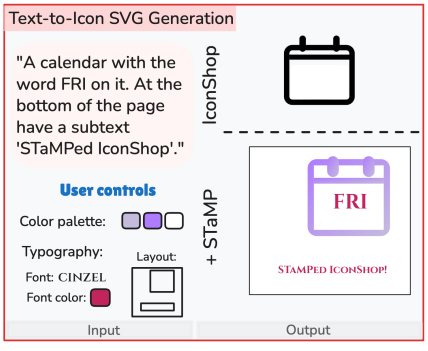

Figure 2: Illustration of STaMP on a monochrome-only backbone (IconShop): decode-time control enables zero-shot color, typography, and layout changes. Baseline outputs remain monochrome; whereas STaMP obeys the palette, arranges layout as specified, and renders the requested text and font.

5.2 End-to-End Design

Experiment setting: Most design models today are raster-first: they produce pixels that are hard to edit procedurally after generation, and practical

Figure 3: Qualitative end-to-end design generation: STaMP-controlled model vs baselines. STaMP yields SVG designs that satisfy color, typography, and layout constraints while maintaining visual coherence and editability; baselines frequently drift from the specified palette, misplace elements, or misrender text. More results in supplementary.

systems stitch multiple components (layout, asset placement, styling, typography) into a pipeline without a single gradient path (Chen et al., 2025). Real design workflows, however, demand precise *controllability* (in form of palette, layout, typography) and *editability* (structured, token-level changes) so that specifications can be met and iterated rapidly. We sidestep the raster bottleneck by treating design as *code*–end-to-end SVGs that explicitly encode geometry, hierarchy, color, and type. While "design-as-code" has been explored in adjacent contexts (Seol et al., 2024), it has not been realized as full text-to-SVG design models. Concretely, we fine-tune two backbones: Qwen3 (8B) and an GPT-OSS (20B), on a proprietary end-to-end SVG design corpus, yielding (to our knowledge) the first text-to-SVG design models. We subsequently stress-test STaMP in this setting, where fidelity, control and editability matter most, to assess how reliably it steers these models under complete briefs. We compare against strong general-purpose code LMs–GPT-5 (OpenAI, 2025), Claude-4 (Anthropic, 2025), Gemini-2.5 (Comanici et al., 2025), Qwen3 (8B) (Yang et al., 2025a), and GPT-OSS (20B) (Agarwal et al., 2025b), covering proprietary and open-source baselines.

Evaluation protocol: Since we claim *text-to-design* SVGs, we first verify design quality and prompt faithfulness with two design-specific metrics: (i) CLIP-Aesthetic (LAION aesthetic predictor, a learned proxy for human visual appeal on rendered designs; and RLR (ROUGE-L Recall) between the prompt and the concatenated text extracted from SVG `<text>` nodes, to check that the design's copy reflects the instruction. We then report the controllability metrics introduced earlier for color, typography and layout.

5.2.1 RQ2 Key Results:

Table 3: Comparison across Unconstrained, Color, Layout, Typography, and C+L+T constraints.

Model	Unconstrained			Color				Layout				Typography				C+L+T					
	CLIP-A ↑	RLR ↑	Complexity (# Tokens)	CIEDE 2000 ↓	CLIP-A ↑	RLR ↑	Complexity (# Tokens)	Overlap % ↑	CLIP-A ↑	RLR ↑	Complexity (# Tokens)	Font Match % ↑	CLIP-A ↑	RLR ↑	Complexity (# Tokens)	CIEDE 2000 ↓	Overlap % ↑	Font Match % ↑	CLIP-A ↑	RLR ↑	Complexity (# Tokens)
GPT 5	4.51	0.88	8.12k	10.32	4.29	0.87	8.06k	34.37	4.16	0.89	8.55k	95.37	4.23	0.87	8.11k	11.18	33.92	95.14	4.35	0.86	8.54k
Claude 4	3.77	0.89	6.97k	11.96	3.48	0.90	7.00k	36.84	3.65	0.92	7.33k	92.76	3.74	0.91	7.02k	12.35	34.26	89.23	3.82	0.89	7.48k
Gemini 2.5	3.96	0.93	7.53k	10.57	3.81	0.91	7.59k	32.61	3.93	0.94	7.95k	90.28	4.05	0.93	7.56k	10.59	30.83	88.15	4.17	0.94	7.94k
Qwen3 (8B)	3.44	0.89	6.04k	12.15	3.43	0.88	6.02k	21.34	3.57	0.85	6.45k	89.19	3.62	0.84	6.06k	13.07	20.31	87.82	3.73	0.83	6.45k
GPT-OSS(20B)	4.11	0.89	7.21k	10.54	4.17	0.91	7.24k	33.95	4.32	0.94	7.63k	85.84	4.46	0.92	7.24k	10.62	33.48	81.16	4.51	0.91	7.82k
Qwen3 (8B)-FT	7.91	0.95	13.84k	3.92	7.68	0.96	13.86k	48.69	7.45	0.93	13.96k	95.63	7.51	0.91	13.88k	4.03	47.75	94.31	7.64	0.90	14.16k
GPT-OSS (20B)-FT	7.12	0.97	14.95k	4.16	6.84	0.91	14.91k	51.38	6.71	0.92	14.89k	95.07	6.85	0.90	14.53k	4.16	51.64	94.75	6.92	0.88	14.99k
Qwen3 (8B)-FT+STaMP	7.91	0.96	13.84k	0.03	7.73	0.97	13.88k	97.27	7.98	0.96	13.96k	100.00	8.03	0.97	13.86k	0.05	98.12	100.00	8.15	0.95	14.24k
GPT-OSS (20B)-FT+STaMP	7.10	0.98	14.94k	0.03	7.26	0.99	14.99k	94.93	7.17	0.95	14.22k	100.00	7.28	0.96	14.53k	0.01	96.28	100.00	7.36	0.97	14.91k

Key result #3: Text-to-SVG fine-tuning yields state-of-the-art design quality, surpassing strong code LMs. On the end-to-end briefs, the fine-tuned text-to-SVG models (Qwen3-8B-FT, GPT-OSS-FT) top the general-purpose code LMs across *both* design metrics–higher CLIP-Aesthetic and higher RLR–consistently across prompts (see Table 3 and qualitative results 3). These gains are substantive: fine-tuning teaches SVG-specific composition (e.g., grouping, coordinate frames, layering of shapes and text), which stabilizes where and how `<text>` nodes are emitted (raising RLR) and yields more balanced, appealing arrangements (raising CLIP-A). As a downstream consequence of that structural competence, control metrics also move sharply in the right direction (lower CIEDE-2000, higher layout overlap, stronger font matches), indicating the model has learned the SVG knobs that make palette, layout, and type editable and precise in a single pass.

Key result #4: STaMP delivers full control on end-to-end designs *without* reducing creative expression. On full briefs (color+layout+typography), the STaMP variants of our fine-tuned text-to-SVG models meet all three controls simultaneously *and* retain the design quality and variety learned during fine-tuning: CLIP-Aesthetic and RLR stay at the fine-tuned baseline or improve, and the qualitative panel shows diverse, on-spec compositions rather than template collapse (see the Table 3 and Fig. 1, 3). Methods without STaMP frequently miss at least one axis under the

same briefs; only the STaMP configurations achieve all the user-defined constraint satisfaction while preserving the backbone's creative range.

5.3 Comparative Control

Experiment setting: To probe the control–quality–efficiency trade-off under identical conditions, we evaluate all methods on the same end-to-end design generation setup as the previous section (identical specs/prompts and decoding harness), using the same two backbones: Qwen-3 (8B) and GPT-OSS (20B). We keep the backbone and prompts fixed so that any change in outcomes reflects only the *control policy* at inference time. We compare controllers in five clusters to expose the trade-offs: (i) *Prompting family*: Prompt-only (vanilla models), Prompt-only (with finetuned models), finetuned + multi-turn prompting: gauges how far instruction following and interactive prompting can go without an explicit controller (with the latency cost of extra turns). (ii) *Test-time guidance/editing*: GeDi-style guidance, ScoPE-style progressive editing: measures generic attribute steering and edits that lack structural guarantees. (iii) *Search/rerank*: Rejection + rerank, Constrained Beam Search (CBS): non-learned optimization via sampling or viability-aware beam, mapping satisfaction versus compute. (iv) *Factorized ablations*: Only Soft Tilt, Only Mask: isolates soft moment steering versus hard feasibility to attribute STaMP's gains. (v) *Ours*: STaMP: a unified Soft Tilt + Mask policy.

Evaluation protocol: All methods consume the same design specifications (color, layout, typography) and decode under matched settings (identical prompts, max length, temperature). For search/rerank baselines, we tune candidate pools and beam widths to their operating points, and report the resulting generation time alongside outcomes. We report the controllability metrics introduced earlier together with CLIP-Aesthetic and generation time. The only additional metric here is well-formedness satisfaction rate (WFSR), the fraction of outputs that parse as valid SVG under a strict parser.

5.3.1 RQ3 Key Results:

Key result #5: STaMP sits on the control–quality–efficiency Pareto front for controlled SVG generation. Across all backbones, STaMP delivers the tightest control–near-zero ΔE_{2000} for color, top-tier layout Overlap, perfect font matching, and 100% well-formedness, while adding only modest generation time compared to heavy search baselines. The ablations reveal the trade-offs: Only Soft Tilt improves color/typography but loses validity; Only Mask secures validity yet leaves soft targets under-optimized; GeDi/ScoPE nudge soft metrics without structural guarantees; CBS and Rejection approach high compliance but incur substantial time costs (and still trail on color or layout in our setting). Notably, STaMP preserves perceptual quality (CLIP-Aesthetic is competitive), indicating that tighter control need not sacrifice aesthetics. In short, the combined Soft Tilt+Mask policy dominates the control–quality–efficiency frontier, delivering simultaneous constraint satisfaction with validity guarantees at single-pass, inference-time cost.

Table 4: Control-method comparison for Qwen3 8B and GPT-OSS-20B under C+L+T constraints.

Model	Control Method	C+L+T Metrics					
		CIEDE 2000 ↓	Overlap % ↑	Font Match % ↑	CLIP-A ↑	WFSR % ↑	Generation time (s) ↓
Qwen-3 (8B)	Prompt (vanilla)	9.84	31.87	61.23	3.73	37.16	154
	Prompt (finetuned)	6.35	39.42	83.69	7.51	61.85	162
	FT+multi-turn Prompt	1.57	43.18	100.00	7.23	69.34	313
	ScoPE	2.63	64.85	91.47	7.01	59.72	176
	GEDi	1.95	69.58	94.26	7.14	62.83	178
	Rejection Sampling	0.64	89.17	100.00	7.30	100.00	486
	CBS	1.09	91.28	100.00	7.12	100.00	247
	Only Soft tilt	0.82	72.19	100.00	7.48	56.31	179
	Only DFA Mask	6.17	36.64	87.86	7.25	100.00	168
	STaMP	0.03	97.15	100.00	6.92	100.00	184
GPT-OSS (20B)	Prompt w/o FT	7.12	46.18	66.84	4.51	41.06	168
	Prompt w/ FT	5.53	50.74	78.95	6.85	61.27	173
	Multi-turn Prompt w/ FT	1.16	56.19	100.00	6.90	63.82	341
	ScoPE	2.56	68.47	92.13	6.92	60.52	187
	GEDi	1.84	72.96	95.31	7.05	64.18	189
	Rejection Sampling	0.57	91.35	100.00	7.25	100.00	512
	CBS	1.14	93.06	100.00	7.18	100.00	262
	Only Soft tilt	0.91	78.17	100.00	6.38	58.94	182
	Only DFA Mask	5.08	47.43	80.15	6.83	100.00	177
	STaMP	0.02	97.18	100.00	7.36	100.00	189

6 Conclusion

We introduce STaMP (Soft Tilt-and-Mask Policy), the first training-free and model-agnostic inference-time decoding controller that enforces structural and semantic constraints during autoregressive SVG generation. By framing control as an I-projection and factorizing it into a soft tilt for probabilistic reweighting and a hard mask for validity, STaMP achieves reliable palette, layout, and typography control across diverse backbones without sacrificing fluency. Beyond enabling workflows such as text-to-design generation of posters and cards, STaMP highlights how principled inference-time interventions can unlock controllable generation without retraining. A key next step is to extend STaMP from local constraint enforcement towards *global*, *higher-order design objectives* (e.g., composition, balance, accessibility), enabling models to satisfy not just token-level rules but holistic design principles during generation.

REFERENCES

Bhavik Agarwal, Ishan Joshi, and Viktoria Rojkova. Think inside the json: Reinforcement strategy for strict llm schema adherence. *arXiv preprint arXiv:2502.14905*, 2025a.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025b.

Lakshya A Agrawal, Aditya Kanade, Navin Goyal, Shuvendu Lahiri, and Sriram Rajamani. Monitor-guided decoding of code lms with static analysis of repository context. *Advances in Neural Information Processing Systems*, 36: 32270–32298, 2023.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Guided open vocabulary image captioning with constrained beam search. *arXiv preprint arXiv:1612.00576*, 2016.

Anthropic. Anthropic claude 4 system card: Claude opus 4 claude sonnet 4. 2025.

Debangshu Banerjee, Tarun Suresh, Shubham Ugare, Sasa Misailovic, and Gagandeep Singh. Crane: Reasoning with constrained llm generation. In *ICLR 2025 Workshop: VerifAI: AI Verification in the Wild*.

Luca Beurer-Kellner, Marc Fischer, and Martin Vechev. Guiding llms the right way: Fast, non-invasive constrained generation. *arXiv preprint arXiv:2403.06988*, 2024.

Nikolay Bogoychev and Pinzhen Chen. Terminology-aware translation with constrained decoding and large language model prompting. *arXiv preprint arXiv:2310.05824*, 2023.

Nan Cao, Xin Yan, Yang Shi, and Chaoran Chen. Ai-sketcher: a deep generative model for producing high-quality sketches. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 2564–2571, 2019.

Alexandre Carlier, Martin Danelljan, Alexandre Alahi, and Radu Timofte. Deepsvg: A hierarchical generative network for vector graphics animation. *Advances in Neural Information Processing Systems*, 33:16351–16361, 2020.

Haoyu Chen, Xiaojie Xu, Wenbo Li, Jingjing Ren, Tian Ye, Songhua Liu, Ying-Cong Chen, Lei Zhu, and Xinchao Wang. Posta: A go-to framework for customized artistic poster generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 28694–28704, 2025.

Sehyun Choi, Tianqing Fang, Zhaowei Wang, and Yangqiu Song. Kcts: Knowledge-constrained tree search decoding with token-level hallucination detection. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14035–14053, 2023.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Imre Csiszár, Paul C Shields, et al. Information theory and statistics: A tutorial. *Foundations and Trends® in Communications and Information Theory*, 1(4):417–528, 2004.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*, 2019.

Jasper Dekoninck, Marc Fischer, Luca Beurer-Kellner, and Martin Vechev. Controlled text generation via language model arithmetic. *arXiv preprint arXiv:2311.14479*, 2023.

Yihong Dong, Xue Jiang, Yuchen Liu, Ge Li, and Zhi Jin. Codepad: Sequence-based code generation with pushdown automaton. *arXiv preprint arXiv:2211.00818*, 2022.

Yixin Dong, Charlie F Ruan, Yaxing Cai, Ruihang Lai, Ziyi Xu, Yilong Zhao, and Tianqi Chen. Xgrammar: Flexible and efficient structured generation engine for large language models. *arXiv preprint arXiv:2411.15100*, 2024.

Jarad Forristal, Fatemehsadat Mireshghallah, Greg Durrett, and Taylor Berg-Kirkpatrick. A block metropolis-hastings sampler for controllable energy-based text generation. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pp. 403–413, 2023.

Kevin Frans, Lisa Soros, and Olaf Witkowski. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *Advances in Neural Information Processing Systems*, 35:5207–5218, 2022.

Saibo Geng, Berkay Döner, Chris Wendler, Martin Josifoski, and Robert West. Sketch-guided constrained decoding for boosting blackbox large language models without logit access. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 234–245, 2024.

Emmanuel Anaya Gonzalez, Sairam Vaidya, Kanghee Park, Ruyi Ji, Taylor Berg-Kirkpatrick, and Loris D'Antoni. Constrained sampling for language models should be easy: An mcmc perspective. *arXiv preprint arXiv:2506.05754*, 2025.

David Ha and Douglas Eck. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*, 2017.

Skyler Hallinan, Alisa Liu, Yejin Choi, and Maarten Sap. Detoxifying text with marco: Controllable revision with experts and anti-experts. *arXiv preprint arXiv:2212.10543*, 2022.

Arthur Hemmer, Mickaël Coustaty, Nicola Bartolo, Jerome Brachat, and Jean-Marc Ogier. Lazy-k decoding: Constrained decoding for information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6727–6736. Association for Computational Linguistics, 2023.

Chris Hokamp and Qun Liu. Lexically constrained decoding for sequence generation using grid beam search. *arXiv preprint arXiv:1704.07138*, 2017.

J Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. Improved lexically constrained decoding for translation and monolingual rewriting. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 839–850, 2019.

Teng Hu, Ran Yi, Baihong Qian, Jiangning Zhang, Paul L Rosin, and Yu-Kun Lai. Supersvg: Superpixel-based scalable vector graphics synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24892–24901, 2024.

Ajay Jain, Amber Xie, and Pieter Abbeel. Vectorfusion: Text-to-svg by abstracting pixel-based diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1911–1920, 2023.

Terry Koo, Frederick Liu, and Luheng He. Automata-based constraints for language model decoding. *arXiv preprint arXiv:2407.08103*, 2024.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. Gedi: Generative discriminator guided sequence generation. *arXiv preprint arXiv:2009.06367*, 2020.

Sachin Kumar, Eric Malmi, Aliaksei Severyn, and Yulia Tsvetkov. Controlled text generation as continuous optimization with multiple constraints. *Advances in Neural Information Processing Systems*, 34:14542–14554, 2021.

Sachin Kumar, Biswajit Paria, and Yulia Tsvetkov. Gradient-based constrained sampling from language models. *arXiv preprint arXiv:2205.12558*, 2022a.

Sachin Kumar, Biswajit Paria, and Yulia Tsvetkov. Gradient-based constrained sampling from language models. *arXiv preprint arXiv:2205.12558*, 2022b.

Duong Minh Le, Yang Chen, Alan Ritter, and Wei Xu. Constrained decoding for cross-lingual label projection. In *The Twelfth International Conference on Learning Representations*.

Alexander K Lew, Tan Zhi-Xuan, Gabriel Grand, and Vikash K Mansinghka. Sequential monte carlo steering of large language models using probabilistic programs. *arXiv preprint arXiv:2306.03081*, 2023.

Zelong Li, Wenyue Hua, Hao Wang, He Zhu, and Yongfeng Zhang. Formal-llm: Integrating formal language and natural language for controllable llm-based agents. *arXiv preprint arXiv:2402.00798*, 2024.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. Commongen: A constrained text generation challenge for generative commonsense reasoning. *arXiv preprint arXiv:1911.03705*, 2019.

Benjamin Lipkin, Benjamin LeBrun, Jacob Hoover Vigly, João Loula, David R MacIver, Li Du, Jason Eisner, Ryan Cotterell, Vikash Mansinghka, Timothy J O'Donnell, et al. Fast controlled generation from language models with adaptive weighted rejection sampling. *arXiv preprint arXiv:2504.05410*, 2025.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A Smith, and Yejin Choi. Dexperts: Decoding-time controlled text generation with experts and anti-experts. *arXiv preprint arXiv:2105.03023*, 2021.

Xin Liu, Muhammad Khalifa, and Lu Wang. Bolt: Fast energy-based controlled text generation with tunable biases. *arXiv preprint arXiv:2305.12018*, 2023.

Raphael Gontijo Lopes, David Ha, Douglas Eck, and Jonathon Shlens. A learned representation for scalable vector graphics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7930–7939, 2019.

Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Neurologic decoding:(un) supervised neural text generation with predicate logic constraints. *arXiv preprint arXiv:2010.12884*, 2020.

Xu Ma, Yuqian Zhou, Xingqian Xu, Bin Sun, Valerii Filev, Nikita Orlov, Yun Fu, and Humphrey Shi. Towards layer-wise image vectorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16314–16323, 2022.

Tao Meng, Sidi Lu, Nanyun Peng, and Kai-Wei Chang. Controllable text generation with neurally-decomposed oracle. *Advances in Neural Information Processing Systems*, 35:28125–28139, 2022.

Fatemehsadat Mireshghallah, Kartik Goyal, and Taylor Berg-Kirkpatrick. Mix and match: Learning-free controllable text generation using energy language models. *arXiv preprint arXiv:2203.13299*, 2022.

Niels Mündler, Jingxuan He, Hao Wang, Koushik Sen, Dawn Song, and Martin Vechev. Type-constrained code generation with language models. *Proceedings of the ACM on Programming Languages*, 9(PLDI):601–626, 2025.

OpenAI. Introducing gpt-5. 2025.

Kanghee Park, Jiayu Wang, Taylor Berg-Kirkpatrick, Nadia Polikarpova, and Loris D'Antoni. Grammar-aligned decoding. *Advances in Neural Information Processing Systems*, 37:24547–24568, 2024.

Kanghee Park, Timothy Zhou, and Loris D'Antoni. Flexible and efficient grammar-constrained decoding. *arXiv preprint arXiv:2502.05111*, 2025.

Gabriel Poesia, Oleksandr Polozov, Vu Le, Ashish Tiwari, Gustavo Soares, Christopher Meek, and Sumit Gulwani. Synchromesh: Reliable code generation from pre-trained language models. *arXiv preprint arXiv:2201.11227*, 2022.

Sagi Polaczek, Yuval Alaluf, Elad Richardson, Yael Vinker, and Daniel Cohen-Or. Neuralsvg: An implicit representation for text-to-vector generation. *arXiv preprint arXiv:2501.03992*, 2025.

Matt Post and David Vilar. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. *arXiv preprint arXiv:1804.06609*, 2018.

Lianhui Qin, Sean Welleck, Daniel Khashabi, and Yejin Choi. Cold decoding: Energy-based constrained text generation with langevin dynamics. *Advances in Neural Information Processing Systems*, 35:9538–9551, 2022.

Pradyumna Reddy, Michael Gharbi, Michal Lukac, and Niloy J Mitra. Im2vec: Synthesizing vector graphics without vector supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7342–7351, 2021.

Leo Sampaio Ferraz Ribeiro, Tu Bui, John Collomosse, and Moacir Ponti. Sketchformer: Transformer-based representation for sketched structure. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14153–14162, 2020.

Juan A Rodriguez, Shubham Agarwal, Issam H Laradji, Pau Rodriguez, David Vazquez, Christopher Pal, and Marco Pedersoli. Starvector: Generating scalable vector graphics code from images. *arXiv preprint arXiv:2312.11556*, 2023.

Peter Schaldenbrand, Zhixuan Liu, and Jean Oh. Styleclipdraw: Coupling content and style in text-to-drawing synthesis. *arXiv preprint arXiv:2111.03133*, 2021.

János Schanda. *Colorimetry: understanding the CIE system*. John Wiley & Sons, 2007.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. Picard: Parsing incrementally for constrained auto-regressive decoding from language models. *arXiv preprint arXiv:2109.05093*, 2021.

Jaejung Seol, Seojun Kim, and Jaejun Yoo. Posterllama: Bridging design ability of language model to content-aware layout generation. In *European Conference on Computer Vision*, pp. 451–468. Springer, 2024.

Jinyue Su, Jiacheng Xu, Xipeng Qiu, and Xuanjing Huang. Incorporating discriminator in sentence generation: a gibbs sampling method. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Xintong Sun, Chi Wei, Minghao Tian, and Shiwen Ni. Earley-driven dynamic pruning for efficient structured decoding. *arXiv preprint arXiv:2506.01151*, 2025.

Vikas Thamizharasan, Difan Liu, Matthew Fisher, Nanxuan Zhao, Evangelos Kalogerakis, and Michal Lukac. Nivel: Neural implicit vector layers for text-to-vector generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4589–4597, 2024.

Shubham Ugare, Rohan Gumaste, Tarun Suresh, Gagandeep Singh, and Sasa Misailovic. Itergen: Iterative semantic-aware structured llm generation with backtracking. *arXiv preprint arXiv:2410.07295*, 2024a.

Shubham Ugare, Tarun Suresh, Hangoo Kang, Sasa Misailovic, and Gagandeep Singh. Syncode: Llm generation with grammar augmentation. *Transactions on Machine Learning Research*, 2024b.

Yael Vinker, Ehsan Pajouheshgar, Jessica Y Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. Clipasso: Semantically-aware object sketching. *ACM Transactions on Graphics (TOG)*, 41(4):1–11, 2022.

Feiyu Wang, Zhiyuan Zhao, Yuandong Liu, Da Zhang, Junyu Gao, Hao Sun, and Xuelong Li. Svgen: Interpretable vector graphics generation with large language models. *arXiv preprint arXiv:2508.09168*, 2025.

Ronghuan Wu, Wanchao Su, Kede Ma, and Jing Liao. Iconshop: Text-guided vector icon synthesis with autoregressive transformers. *ACM Transactions on Graphics (TOG)*, 42(6):1–14, 2023.

Ronghuan Wu, Wanchao Su, and Jing Liao. Chat2svg: Vector graphics generation with large language models and image diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 23690–23700, 2025.

Ximing Xing, Chuang Wang, Haitao Zhou, Jing Zhang, Qian Yu, and Dong Xu. Diffsketcher: Text guided vector sketch synthesis through latent diffusion models. *Advances in Neural Information Processing Systems*, 36:15869–15889, 2023.

Ximing Xing, Haitao Zhou, Chuang Wang, Jing Zhang, Dong Xu, and Qian Yu. Svgdreamer: Text guided svg generation with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4546–4555, 2024.

Ximing Xing, Juncheng Hu, Guotao Liang, Jing Zhang, Dong Xu, and Qian Yu. Empowering llms to understand and generate complex vector graphics. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 19487–19497, 2025.

Kota Yamaguchi. Canvasvae: Learning to generate vector graphic documents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5481–5489, 2021.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Kevin Yang and Dan Klein. Fudge: Controlled text generation with future discriminators. *arXiv preprint arXiv:2104.05218*, 2021.

Yiying Yang, Wei Cheng, Sijin Chen, Xianfang Zeng, Fukun Yin, Jiaxu Zhang, Liao Wang, Gang Yu, Xingjun Ma, and Yu-Gang Jiang. Omnisvg: A unified scalable vector graphics generation model. *arXiv preprint arXiv:2504.06263*, 2025b.

Haotian Ye, Himanshu Jain, Chong You, Ananda Theertha Suresh, Haowei Lin, James Zou, and Felix Yu. Efficient and asymptotically unbiased constrained decoding for large language models. *arXiv preprint arXiv:2504.09135*, 2025.

Sangwon Yu, Changmin Lee, Hojin Lee, and Sungroh Yoon. Controlled text generation for black-box language models via score-based progressive editor. *arXiv preprint arXiv:2311.07430*, 2023.

Honghua Zhang, Meihua Dang, Nanyun Peng, and Guy Van den Broeck. Tractable control for autoregressive language generation. In *International Conference on Machine Learning*, pp. 40932–40945. PMLR, 2023.

Peiying Zhang, Nanxuan Zhao, and Jing Liao. Text-to-vector generation with neural path representation. *ACM Transactions on Graphics (TOG)*, 43(4):1–13, 2024.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Tianqi Zhong, Quan Wang, Jingxuan Han, Yongdong Zhang, and Zhendong Mao. Air-decoding: Attribute distribution reconstruction for decoding-time controllable text generation. *arXiv preprint arXiv:2310.14892*, 2023.

A THEORETICAL GUARANTEES OF STAMP

A.1 HARD-VALIDITY (SOUNDNESS) OF STAMP

Setting. Recall the feasible set:

$$\mathcal{C} \;=\; \{\, x \in \mathcal{V}^* :\; \mathcal{R} \text{ accepts } \mathrm{SVG}(x) \,\},$$

where $\mathcal{R}$ is a deterministic recognizer for well-formed SVG, and the one-step policy produced by STaMP,

$$P_{\text{tilt+mask}}(x_t \mid x_{<t}) \;\propto\; P_{\text{tilt}}(x_t \mid x_{<t})\, \mathbf{1}_{\{x_t \in \mathcal{V}_t\}}, \qquad P_{\text{tilt}}(x_t \mid x_{<t}) \propto P_\theta(x_t \mid x_{<t})\, \exp\big(b_\phi(x_{<t}, x_t)\big), \tag{11}$$

with $\mathcal{V}_t$ the set of viable tokens at prefix $x_{<t}$. Let s_t denote the state of $\mathcal{R}$ after feeding the decoded prefix $\mathrm{SVG}(x_{<t})$; write δ for the (deterministic) state transition function of $\mathcal{R}$ composed with the token decoder, i.e., $\delta(s_t, x_t)$ is the state reached after appending token x_t (expanding it into characters via dec and streaming them through $\mathcal{R}$). We assume exact viability:

$$\mathcal{V}_t \;=\; \big\{\, v \in \mathcal{V} :\; \exists\, y \in \mathcal{V}^* \text{ s.t. } \delta\big(\delta(s_t, v), y\big) \in F \text{ and the termination rule accepts} \,\big\}, \tag{12}$$

where F is the set of accepting states of $\mathcal{R}$ and the termination rule is the one used in the main text (EOS-terminated sequences, or fixed-length sequences). In the EOS case we additionally gate EOS: the EOS token is included in $\mathcal{V}_t$ if and only if $s_t \in F$.

Claim (Soundness). Let $\tilde{Q}_\phi$ be the sequence distribution induced by sampling $x_t \sim P_{\text{tilt+mask}}(\cdot \mid x_{<t})$ at each step and terminating only when EOS is sampled (EOS-terminated) or when the prescribed fixed length is reached (fixed-length). Under the exact-viability specification equation 12 (and EOS gating when applicable), every realized sequence lies in the feasible set:

$$\tilde{Q}_\phi(\mathcal{C}) \;=\; 1.$$

Proof. We give the argument for the EOS-terminated case; the fixed-length case is identical after replacing the EOS gate by a length budget in the viability test.

We prove by induction on t the invariant:

$$\mathsf{Inv}(t): \quad \text{(i) } s_t \text{ is the state reached by } \mathrm{SVG}(x_{<t}), \quad \text{(ii) there exists a valid completion } y \in \mathcal{V}^* \text{ with } \delta(s_t, y) \in F.$$

Base case. At $t = 1$, $x_{<1}$ is empty, s_1 is the start state of $\mathcal{R}$. Feasibility of the task ensures there exists some valid $x \in \mathcal{C}$; hence (ii) holds for $y = x$. Thus $\mathsf{Inv}(1)$ holds.

Inductive step. Assume $\mathsf{Inv}(t)$ holds. By exact viability equation 12, the mask in equation 11 restricts sampling to $\mathcal{V}_t$, i.e., to tokens x_t for which there *exists* a suffix y with $\delta(\delta(s_t, x_t), y) \in F$. The controller samples some $x_t \in \mathcal{V}_t$ with nonzero probability, then updates the recognizer state to:

$$s_{t+1} \;=\; \delta(s_t, x_t).$$

By the defining property of $\mathcal{V}_t$, there is at least one y such that $\delta(s_{t+1}, y) \in F$, so (ii) holds at $t + 1$; (i) holds by construction. Hence $\mathsf{Inv}(t + 1)$.

Termination and acceptance. In the EOS-terminated setting, the process halts only when EOS is sampled. By EOS gating, EOS $\in \mathcal{V}_t$ if and only if $s_t \in F$. Thus, the only way to terminate is from an accepting state. (For every reachable accepting state, the conditional probability of emitting EOS under $P_{\text{tilt+mask}}(\text{EOS}|\cdot)$ is bounded. Therefore, standard geometric-series arguments imply the process terminates almost surely and with finite expectation). The realized sequence x satisfies $\mathcal{R}$ accepts $\mathrm{SVG}(x)$, i.e., $x \in \mathcal{C}$. Therefore $\tilde{Q}_\phi(\mathcal{C}) = 1$.

Fixed length. When a fixed length T is imposed, the viability definition equation 12 is understood with a budget on the remaining steps: a token v is viable at time t only if there exists y of length at most $T - t$ such that $\delta(\delta(s_t, v), y) \in F$. The inductive invariant carries through verbatim, and at $t = T$ only prefixes that admit an accepting completion of length 0 (i.e., already accepting) can occur. Hence the realized sequence is accepted by $\mathcal{R}$ and lies in $\mathcal{C}$.

Discussion. The guarantee is *soundness*, not *completeness*. Soundness says every sequence the controller emits is in $\mathcal{C}$; it does not claim that every $x \in \mathcal{C}$ is reachable with positive probability. Completeness would additionally require liveness/progress conditions (Appendix A.7) ensuring the viable set never empties along trajectories. For the hard-validity contract, soundness is the essential safety property: regardless of the learned Soft component, as long as the Mask enforces exact viability and EOS gating, the controller cannot produce an invalid SVG.

A.2 MASK-ONLY ≡ CONDITIONING (WHEN $\eta = 0$)

Setting. Consider the hard constraint set:

$$\mathcal{C} \;=\; \{\, x \in \mathcal{V}^* :\; \mathcal{R} \text{ accepts } \mathrm{SVG}(x) \,\},$$

the base autoregressive model P_θ as in equation 1–equation 2, and the Mask policy equation 10. In this appendix we set $\eta = 0$ (no Soft Tilt), i.e., $b_\phi \equiv 0$ and $P_{\text{tilt}}(\cdot \mid x_{<t}) = P_\theta(\cdot \mid x_{<t})$. For any prefix $x_{<t}$ with $P_\theta(x_{<t}) > 0$, define the base-model acceptance probability:

$$G_0(x_{<t}) \;:=\; E_{x_{t:}\sim P_\theta}\big[\mathbf{1}_{\{x\in\mathcal{C}\}} \;\big|\; x_{<t}\big] \;=\; P_\theta\big(x \in \mathcal{C} \;\big|\; x_{<t}\big), \tag{13}$$

and similarly $G_0(x_{<t}v) = P_\theta(x \in \mathcal{C} \mid x_{<t}v)$. For the equivalence below we use the P_θ-reachable viable set:

$$\mathcal{V}_t \;=\; \{\, v \in \mathcal{V} :\; G_0(x_{<t}v) > 0 \,\},$$

which refines recognizer viability by excluding next tokens that admit accepting completions only with zero P_θ-probability. In the EOS-terminated setting, EOS is allowed iff the current recognizer state is accepting (equivalently, $G_0(x_{<t}\langle\text{eos}\rangle) > 0$). The Mask-only next-token policy is then:

$$\pi_{\text{mask}}(x_t \mid x_{<t}) \;=\; \frac{P_\theta(x_t \mid x_{<t})\,\mathbf{1}_{\{x_t\in\mathcal{V}_t\}}}{\sum_{v\in\mathcal{V}_t} P_\theta(v \mid x_{<t})}. \tag{14}$$

Target conditional under the base model. Conditioning P_θ on eventual validity defines:

$$P_\theta\big(x_t \mid x_{<t},\, x \in \mathcal{C}\big) \;=\; \frac{P_\theta\big(x_t,\, x \in \mathcal{C} \mid x_{<t}\big)}{P_\theta\big(x \in \mathcal{C} \mid x_{<t}\big)} \;=\; \frac{P_\theta(x_t \mid x_{<t})\, G_0(x_{<t}x_t)}{G_0(x_{<t})}, \tag{15}$$

where the last equality is the law of total probability over suffixes under P_θ.

Claim (Equivalence, with necessary and sufficient condition). Fix any prefix $x_{<t}$ with $G_0(x_{<t}) > 0$. Then

$$\pi_{\text{mask}}(\cdot \mid x_{<t}) \;=\; P_\theta(\cdot \mid x_{<t},\, x \in \mathcal{C}) \quad\Longleftrightarrow\quad G_0(x_{<t}v) \text{ is constant over } v \in \mathcal{V}_t.$$

In particular, if $G_0(x_{<t}v)$ does not depend on the choice of viable next token, then Mask-only sampling coincides with conditioning the base model on eventual validity. If G_0 varies across viable tokens, the two next-token distributions generally differ.

Proof. By equation 14, for any $x_t \in \mathcal{V}_t$,

$$\pi_{\text{mask}}(x_t \mid x_{<t}) \;=\; \frac{P_\theta(x_t \mid x_{<t})}{\sum_{v\in\mathcal{V}_t} P_\theta(v \mid x_{<t})}.$$

By equation 15, for any $x_t \in \mathcal{V}_t$,

$$P_\theta(x_t \mid x_{<t},\, x \in \mathcal{C}) \;=\; \frac{P_\theta(x_t \mid x_{<t})\, G_0(x_{<t}x_t)}{G_0(x_{<t})}, \qquad G_0(x_{<t}) \;=\; \sum_{v\in\mathcal{V}_t} P_\theta(v \mid x_{<t})\, G_0(x_{<t}v),$$

(the last identity sums over all tokens; terms with $G_0 = 0$ vanish and can be dropped). Hence,

$$P_\theta(x_t \mid x_{<t},\, x \in \mathcal{C}) \;=\; \frac{P_\theta(x_t \mid x_{<t})\, G_0(x_{<t}x_t)}{\sum_{v\in\mathcal{V}_t} P_\theta(v \mid x_{<t})\, G_0(x_{<t}v)}.$$

Comparing with π_{mask}, equality for all $x_t \in \mathcal{V}_t$ holds iff $G_0(x_{<t}v)$ is the same constant for all $v \in \mathcal{V}_t$ (so the common factor cancels). This condition is also necessary.

Discussion and implications. Equation equation 15 is the $\eta = 0$ instance of equation 8 and shows that conditioning multiplies base next-token probabilities by $G_0(x_{<t}x_t)$ before renormalization. The Mask-only policy equation 14 enforces $G_0(x_{<t}x_t) > 0$ but omits this multiplicative factor. Consequently, Mask-only is exactly $P_\theta(\cdot \mid x \in \mathcal{C})$ if and only if the acceptance probability is token-invariant across the viable set at that prefix—that is, when all viable next tokens yield the same G_0 under P_θ. In general, especially for long-range constraints, G_0 varies with v and the two policies differ.

From the STaMP viewpoint, Mask enforces the hard requirement (no invalid paths), while the missing multiplicative factor is the $\eta \to 0$ case of the ideal log-bias $b^*(x_{<t}, x_t) = \log G_\eta(x_{<t}x_t) - \log G_\eta(x_{<t})$ in equation 8. Thus, if $b_\phi = b^*$ at $\eta = 0$, the combined Soft Tilt+Mask reproduces conditioning exactly; if $b_\phi \equiv 0$, Mask-only matches conditioning precisely in the token-invariant case above. In all cases, Appendix A.1 applies: regardless of these weights, every realized sequence lies in $\mathcal{C}$ (soundness).

A.3 Existence & Uniqueness of the I-Projection

Setting and assumptions. We study equation 3 over the simplex $\Delta(\mathcal{V}^*)$ with the hard support restriction $Q(\mathcal{C}) = 1$ and soft moment constraint $E_Q[f] = c$ (elementwise when $m > 1$). We adopt the feasibility assumption from the main text: $P_\theta(\mathcal{C}) > 0$ and $c \in \mathcal{M}$, where:

$$\mathcal{M} \;=\; \big\{\, E_Q[f] :\; Q \ll P_\theta,\;\; Q(\mathcal{C}) = 1 \,\big\}.$$

To avoid technicalities unrelated to SVG control, assume either that f is bounded or, more generally, that the exponential moment $\sum_{x\in\mathcal{C}} P_\theta(x)\exp(\eta^\top f(x))$ is finite in a neighborhood of the origin, so the log-partition $\psi(\eta) = \log Z(\eta)$ in equation 5 is well-defined and smooth on its effective domain ($Z(\eta) < \infty \forall \eta$ in an open convex set containing the optimum). These assumptions ensure uniform integrability of f on the feasible slice and continuity of $Q \mapsto E_Q[f]$ under weak convergence.

Existence. Consider the Lagrangian equation 4 with multipliers $\eta \in \mathbb{R}^m$ and $\mu \in \mathbb{R}$, restricted to distributions supported on $\mathcal{C}$. Eliminating Q by the stationarity condition yields the tilted family:

$$Q_\eta(x) \;=\; \frac{1}{Z(\eta)}\, P_\theta(x)\, \exp\big(\eta^\top f(x)\big)\, \mathbf{1}_{\{x\in\mathcal{C}\}}, \qquad Z(\eta) \;=\; \sum_{x\in\mathcal{C}} P_\theta(x) \exp\big(\eta^\top f(x)\big),$$

and the concave dual $g(\eta) = \eta^\top c - \psi(\eta)$ with $\nabla g(\eta) = c - E_{Q_\eta}[f]$ and $\nabla^2 g(\eta) = -\mathrm{Cov}_{Q_\eta}(f) \preceq 0$ (cf. equation 6–equation 7). There are two complementary existence routes. (i) *Dual route.* If c lies in the relative interior of the exponential-family moment image:

$$\mathcal{M}_{\exp} \;=\; \big\{\, E_{Q_\eta}[f] :\; Z(\eta) < \infty \,\big\} \;=\; \mathrm{Im}(\nabla\psi),$$

then by continuity of $\nabla\psi$ there exists η^* with $E_{Q_{\eta^*}}[f] = c$; strong duality gives $Q^* = Q_{\eta^*}$. (ii) *Direct route.* On a countable alphabet the feasible set

$$\{\, Q \in \Delta(\mathcal{V}^*) :\; Q(\mathcal{C}) = 1,\; E_Q[f] = c,\; Q \ll P_\theta \,\}$$

is convex and closed (linearity of constraints and absolute continuity are closed conditions; continuity of $Q \mapsto E_Q[f]$ (Weierstrass) follows from the regularity above). Since $Q \mapsto \mathrm{KL}(Q\|P_\theta)$ is lower semicontinuous and takes values in $[0, \infty]$, the infimum is attained. Boundary cases (see below) are handled by closure of the feasible set.

Uniqueness of the primal optimizer. The mapping $Q \mapsto \mathrm{KL}(Q\|P_\theta)$ is strictly convex in Q on the affine slice defined by the constraints (with $Q \ll P_\theta$). Hence there is at most one minimizer; together with existence, this yields a unique optimizer Q^*, independent of parameterization.

Characterization and dual optimality. By the KKT conditions for equation 4, the unique primal optimizer has the exponential-tilt form:

$$Q^*(x) \;=\; \frac{1}{Z(\eta^*)}\, P_\theta(x)\, \exp\big(\eta^{*\top} f(x)\big)\, \mathbf{1}_{\{x\in\mathcal{C}\}}$$

for some dual maximizer η^*, and satisfies the moment-matching condition $E_{Q^*}[f] = c$. Conversely, any η with $E_{Q_\eta}[f] = c$ yields a feasible Q_η attaining the primal optimum. The Pythagorean identity (Csiszár et al., 2004) for the I-projection holds: for any feasible Q,

$$\mathrm{KL}(Q\|P_\theta) \;=\; \mathrm{KL}(Q\|Q^*) \;+\; \mathrm{KL}(Q^*\|P_\theta),$$

which certifies optimality of Q^* and shows that deviations from Q^* strictly increase the objective.

Uniqueness of multipliers and redundancy. While Q^* is unique, the dual vector η^* is unique iff the features in f are nonredundant under Q^*, e.g., $\mathrm{Cov}_{Q^*}(f) \succ 0$ (equivalently, there is no nonzero $a \in \mathbb{R}^m$ with $a^\top f(x)$ Q^*-a.s. constant). If such redundancy exists (e.g., including a constant feature), the set of dual maximizers is an affine translate along redundant directions; all such η produce the same Q^* because the induced tilt differs only by a multiplicative constant absorbed into $Z(\eta)$.

Boundary cases. If c lies on the boundary of the achievable moment set $\mathcal{M}$, a maximizing sequence η_k may diverge while Q_{η_k} converges (in distribution) to a limit supported on $\mathcal{C}$ that satisfies the constraint; then the primal optimizer still exists and is unique, while the dual optimum is attained only in the extended sense (at infinity). (In other words, if c lies on $\partial\mathcal{M}$, there exists a diverging sequence η_k with $||\eta_k|| \to \infty$, such that Q_{η_k} converges weakly to Q^* - the dual supremum is attained only in the extended sense.) Under the standing regularity (bounded f or finite exponential moments) and c in the relative interior of $\mathcal{M}$, both primal and dual optima are attained with finite η^*.

Discussion. Under the feasibility and regularity conditions above, the I-projection equation 3 admits a unique solution Q^*, realized by an exponential reweighting of P_θ restricted to $\mathcal{C}$, with multipliers chosen to satisfy $E_{Q^*}[f] = c$. This establishes that the target controlled generator is well-posed and reproducible; all approximation error in subsequent sections (e.g., Soft Tilt and Mask) can be interpreted relative to this uniquely defined information-theoretic optimum.

A.4 SOFT-MOMENT CALIBRATION (EXISTENCE / UNIQUENESS / CONVERGENCE)

Setting. The hard constraint is encoded by $\mathcal{C}$ as before, and the soft target is $c \in \mathbb{R}^m$ for a utility $f : \mathcal{V}^* \to \mathbb{R}^m$. For any multiplier $\eta \in \mathbb{R}^m$ in the natural parameter domain $\operatorname{dom}\psi := \{\eta : Z(\eta) < \infty\}$, define:

$$Q_\eta(x) \;=\; \frac{1}{Z(\eta)}\, P_\theta(x)\, \exp\big(\eta^\top f(x)\big)\, \mathbf{1}_{\{x\in\mathcal{C}\}}, \qquad Z(\eta) \;=\; \sum_{x\in\mathcal{C}} P_\theta(x)\, \exp\big(\eta^\top f(x)\big),$$

and $\psi(\eta) = \log Z(\eta)$. As established in equation 6, $\nabla\psi(\eta) = E_{Q_\eta}[f(x)]$ and $\nabla^2\psi(\eta) = \mathrm{Cov}_{Q_\eta}(f) \succeq 0$ (all gradients/Hessians elementwise). The dual objective in equation 7 is $D(\eta) = \eta^\top c - \psi(\eta)$ with gradient $\nabla D(\eta) = c - E_{Q_\eta}[f(x)]$ and Hessian $\nabla^2 D(\eta) = -\mathrm{Cov}_{Q_\eta}(f) \preceq 0$.

Claim (existence). Let,

$$\mathcal{M} \;=\; \Big\{ E_Q[f] :\; Q \ll P_\theta,\; Q(\mathcal{C}) = 1 \Big\} \quad \text{(achievable moments under } P_\theta \text{ on } \mathcal{C}\text{)}.$$

If $c \in \mathrm{ri}(\mathcal{M})$ (relative interior), then there exists $\eta^* \in \operatorname{dom}\psi$ such that $E_{Q_{\eta^*}}[f] = c$, i.e., $D(\eta)$ attains its maximum at a finite η^* and $\nabla D(\eta^*) = 0$. If $c \in \overline{\mathcal{M}} \setminus \mathrm{ri}(\mathcal{M})$ (boundary case), there exists a maximizing sequence η_k with $\|\eta_k\| \to \infty$ such that Q_{η_k} converges (in distribution) to the unique primal optimizer Q^* with $E_{Q^*}[f] = c$.

Reasoning. ψ is convex and lower semicontinuous on $\operatorname{dom}\psi$ (an open convex set), hence D is concave and upper semicontinuous. For $c \in \mathrm{ri}(\mathcal{M})$, standard convex duality implies dual attainment at a finite η^* with first-order optimality $\nabla D(\eta^*) = 0$, i.e., $E_{Q_{\eta^*}}[f] = c$. If c lies on the boundary, the supremum of D is achieved only in the limit; the corresponding Q_{η_k} converges to the primal Q^* that attains equation 3.

Claim (uniqueness). If f is nondegenerate in the sense that $\mathrm{Cov}_{Q_\eta}(f) \succ 0$ in a neighborhood of the solution, then ψ is strictly convex and D is strictly concave; the maximizer η^* is unique, and thus Q_{η^*} is unique. Even if $\mathrm{Cov}_{Q_\eta}(f)$ is only positive semidefinite (e.g., f contains an affine redundancy), the primal optimizer Q^* of equation 3 is still unique because $\mathrm{KL}(Q\|P_\theta)$ is strictly convex in Q over the affine constraint set.

Reasoning. Strict convexity of ψ makes $\nabla\psi$ injective, so $\nabla\psi(\eta) = c$ has at most one (finite) solution. In the degenerate case, multiple multipliers can map to the same Q_η; strict convexity of the primal objective then pins down a unique Q^* even if η is not unique.

Monotonicity and the $m = 1$ specialization. When $m = 1$, $\psi'(\eta) = E_{Q_\eta}[f]$ and $\psi''(\eta) = \mathrm{Var}_{Q_\eta}(f) \geq 0$. If f is not a.s. constant under Q_η in a neighborhood of the solution, then $\psi''(\eta) > 0$ there and $\eta \mapsto E_{Q_\eta}[f]$ is strictly increasing. Consequently, a one-dimensional root-finder (bisection, or Newton with line search) finds η^* robustly.

Convergence of calibration via the dual. Consider gradient ascent on D:

$$\eta_{k+1} \;=\; \eta_k \;+\; \alpha_k \big(c - E_{Q_{\eta_k}}[f]\big),$$

with either a backtracking line search guaranteeing ascent of D, or diminishing stepsizes (α_k) that satisfy $\sum_k \alpha_k = \infty$ and $\sum_k \alpha_k^2 < \infty$. Since D is concave with $\nabla D(\eta) = c - E_{Q_\eta}[f]$ and $\nabla^2 D(\eta) = -\mathrm{Cov}_{Q_\eta}(f) \preceq 0$, the iteration converges to the (unique) maximizer η^* when it exists at finite norm. On compact level sets where ∇D is Lipschitz, fixed stepsizes smaller than the inverse Lipschitz constant also yield global convergence. In the $m = 1$ case, monotonicity of $E_{Q_\eta}[f]$ implies that bracketing with bisection converges linearly to η^*.

Stochastic calibration with Monte Carlo estimates. In practice, $E_{Q_{\eta_k}}[f]$ is estimated from samples. If the estimator is unbiased (or asymptotically unbiased) with bounded variance and (α_k) is a Robbins–Monro stepsize sequence, the stochastic approximation:

$$\eta_{k+1} \;=\; \eta_k \;+\; \alpha_k \big(c - \widehat{E}_{Q_{\eta_k}}[f]\big)$$

converges almost surely to η^* under standard SA conditions (e.g., local strong concavity of D via $\lambda_{\min}(\mathrm{Cov}_{Q_{\eta^*}}(f)) > 0$ and Lipschitz continuity of the mean map). If one replaces Q_{η_k} by the controlled sampler $\tilde{Q}_\phi$ from equation 9–equation 10, and the resulting estimator has a uniform bias bound $\sup_\eta \|E_{\tilde{Q}_\phi}[f] - E_{Q_\eta}[f]\| \leq B$, then the iterates converge to a neighborhood of η^* whose radius scales as $O(B/\mu)$, where μ is the local strong-concavity modulus of D (i.e., the minimal eigenvalue of $\mathrm{Cov}_{Q_{\eta^*}}(f)$). In the oracle limit (ideal b_ϕ and exact masking), $B = 0$ and the iterates converge to η^*.

Under the standing feasibility assumption and mild regularity, there exists a multiplier that achieves the soft target: if $c \in \mathrm{ri}(\mathcal{M})$, a finite η^* satisfies $E_{Q_{\eta^*}}[f] = c$; on the boundary, a maximizing sequence η_k yields $Q_{\eta_k} \Rightarrow Q^*$ with the correct moment. The solution is unique at the distribution level, and dual-gradient calibration converges to η^* (or to the boundary in the limit), with stochastic and approximate variants converging to a quantifiable neighborhood governed by estimator bias and local curvature.

A.5 Chain-Rule KL Decomposition (Token-Level Surrogate is Exact)

Setting. Let $\tilde{Q}_\phi$ be the sequence distribution induced by the controlled decoder (equation 9–equation 10) with EOS termination (almost surely). Thus, for any realized sequence $x = (x_1, \ldots, x_{|x|})$ (terminated by $\langle \text{eos} \rangle$),

$$\tilde{Q}_\phi(x) \;=\; \prod_{t=1}^{|x|} \pi_\phi(x_t \mid x_{<t}), \qquad P_\theta(x) \;=\; \prod_{t=1}^{|x|} P_\theta(x_t \mid x_{<t}),$$

where $\pi_\phi(\cdot \mid x_{<t})$ is the normalized next-token policy after Soft Tilt and Mask, and $P_\theta(\cdot \mid x_{<t})$ is the base next-token distribution from equation 1–equation 2. We assume absolute continuity of $\tilde{Q}_\phi$ with respect to P_θ at the sequence level, which is ensured by the per-step support inclusion:

$$\operatorname{supp} \pi_\phi(\cdot \mid x_{<t}) \;\subseteq\; \operatorname{supp} P_\theta(\cdot \mid x_{<t}) \quad \tilde{Q}_\phi\text{-a.s.}$$

(reweighting preserves zeros and Mask only removes mass). Let h_t denote the decoder state at step t (a measurable function of $x_{<t}$ and the cached history $\mathcal{H}_t$). For brevity we write $\pi_\phi(\cdot \mid h_t)$ and $P_\theta(\cdot \mid h_t)$ in place of conditioning on $x_{<t}$.

Claim (chain-rule KL). If $\mathrm{KL}(\tilde{Q}_\phi \,\|\, P_\theta) < \infty$, then:

$$\mathrm{KL}\Big(\tilde{Q}_\phi \,\|\, P_\theta\Big) \;=\; \sum_{t \geq 1} E_{h_t \sim \tilde{Q}_\phi}\big[\,\mathrm{KL}\big(\pi_\phi(\cdot \mid h_t) \,\|\, P_\theta(\cdot \mid h_t)\big)\big]\,. \tag{16}$$

Proof. By the autoregressive factorization,

$$\log \frac{\tilde{Q}_\phi(x)}{P_\theta(x)} \;=\; \sum_{t=1}^{|x|} \log \frac{\pi_\phi(x_t \mid x_{<t})}{P_\theta(x_t \mid x_{<t})}.$$

Taking expectation under $\tilde{Q}_\phi$ and conditioning on $X_{<t}$ (or h_t) yields:

$$E_{\tilde{Q}_\phi}\left[\log \frac{\tilde{Q}_\phi(X)}{P_\theta(X)}\right] = \sum_{t \geq 1} E_{h_t \sim \tilde{Q}_\phi}\left[E_{X_t \sim \pi_\phi(\cdot \mid h_t)}\left(\log \frac{\pi_\phi(X_t \mid h_t)}{P_\theta(X_t \mid h_t)}\right)\right].$$

For each t, the inner expectation is the discrete KL divergence $\mathrm{KL}\big(\pi_\phi(\cdot \mid h_t) \,\|\, P_\theta(\cdot \mid h_t)\big) \geq 0$. Because these terms are nonnegative, we may exchange the (a.s. finite) sum and the outer expectation by monotone convergence, obtaining equation 16. Finally, by padding sequences after EOS with an absorbing token for which both policies put probability 1, all summands for $t > |X|$ are 0 a.s., so the series is well defined.

Discussion. Equation equation 16 shows that the token-level KL regularizer used in the training objective of § STaMP, namely $\sum_t E_{h_t \sim \tilde{Q}_\phi}\big[\mathrm{KL}(\pi_\phi(\cdot \mid h_t) \,\|\, P_\theta(\cdot \mid h_t))\big]$–is exactly the sequence-level divergence $\mathrm{KL}(\tilde{Q}_\phi \| P_\theta)$ for EOS-terminated autoregressive decoding. Thus the engineering surrogate is not an approximation: it is the chain-rule decomposition of the global information change from the base model. In particular, controlling the per-step KL budget directly controls the overall deviation from P_θ, and by Pinsker's inequality this also bounds the total-variation shift between the induced sequence distributions. The argument is unaffected by the Mask: masking only removes mass (and, together with reweighting, never creates mass where P_θ has none), so absolute continuity and the per-step conditional KL remain well defined at each prefix.

A.6 Oracle Consistency (Exactness if Components are Exact)

Setting. Recall the optimal next-token conditional from equation 8:

$$Q^*(x_t \mid x_{<t}) \;=\; P_\theta(x_t \mid x_{<t}) \cdot \frac{G_\eta(x_{<t}x_t)}{G_\eta(x_{<t})}, \qquad G_\eta(x_{<t}) = E_{x_{t:} \sim P_\theta}\Big[\exp\big(\eta^\top f(x)\big)\, \mathbf{1}_{\{x \in \mathcal{C}\}} \;\Big|\; x_{<t}\Big].$$

The STaMP controller forms at each step the Soft Tilt policy equation 9,

$$P_{\text{tilt}}(x_t \mid x_{<t}) \;\propto\; P_\theta(x_t \mid x_{<t})\, \exp\big(b_\phi(x_{<t}, x_t)\big),$$

followed by the Mask equation 10,

$$P_{\text{tilt+mask}}(x_t \mid x_{<t}) \;\propto\; P_{\text{tilt}}(x_t \mid x_{<t})\, \mathbf{1}_{\{x_t \in \mathcal{V}_t\}}, \qquad \mathcal{V}_t = \{v \in \mathcal{V} :\ \mathcal{R} \text{ remains viable on } x_{<t}v\},$$

where viability includes EOS/length budgets as in Appendix A.1. The ideal log-bias from equation 8 is:

$$b^*(x_{<t}, x_t) \;=\; \log G_\eta(x_{<t}x_t) - \log G_\eta(x_{<t}).$$

Claim (oracle consistency). Suppose (i) the adapter is oracle-accurate, $b_\phi(x_{<t}, x_t) = b^*(x_{<t}, x_t)$ for all prefixes and tokens with positive base support, and (ii) masking enforces exact P_θ-reachable viability, i.e.,

$$x_t \in \mathcal{V}_t \quad\Longleftrightarrow\quad P_\theta\big(\exists\,\text{accepting continuation} \;\big|\; x_{<t}x_t\big) > 0 \quad\Longleftrightarrow\quad G_\eta(x_{<t}x_t) > 0.$$

Then for every prefix $x_{<t}$ with $Q^*(x_{<t}) > 0$, the next-token policies coincide:

$$P_{\text{tilt+mask}}(\cdot \mid x_{<t}) \;=\; Q^*(\cdot \mid x_{<t}).$$

Consequently, the induced sequence distribution of the controlled decoder equals the I-projection optimum:

$$\tilde{Q}_\phi \;=\; Q^*.$$

Proof. Fix a prefix $x_{<t}$ with $Q^*(x_{<t}) > 0$. Then $G_\eta(x_{<t}) > 0$ and $P_\theta(x_{<t}) > 0$. With $b_\phi = b^*$,

$$P_{\text{tilt}}(x_t \mid x_{<t}) \;\propto\; P_\theta(x_t \mid x_{<t})\,\exp\big(\log G_\eta(x_{<t}x_t) - \log G_\eta(x_{<t})\big) \;=\; P_\theta(x_t \mid x_{<t})\,\frac{G_\eta(x_{<t}x_t)}{G_\eta(x_{<t})}.$$

Because $\exp(\eta^\top f)$ is strictly positive, $G_\eta(x_{<t}x_t) = 0$ holds iff $P_\theta(x \in \mathcal{C} \mid x_{<t}x_t) = 0$, i.e., there is no accepting continuation with positive P_θ-probability. By assumption (ii),

$$x_t \in \mathcal{V}_t \quad\Longleftrightarrow\quad G_\eta(x_{<t}x_t) > 0.$$

Applying the Mask multiplies by $\mathbf{1}_{\{G_\eta(x_{<t}x_t)>0\}}$, which leaves the expression unchanged for viable tokens and zeroes it for nonviable ones. Renormalizing over $x_t \in \mathcal{V}_t$ gives:

$$P_{\text{tilt+mask}}(x_t \mid x_{<t}) \;=\; \frac{P_\theta(x_t \mid x_{<t})\,G_\eta(x_{<t}x_t)}{\sum_{v\in\mathcal{V}_t} P_\theta(v \mid x_{<t})\,G_\eta(x_{<t}v)}\,.$$

Using

$$G_\eta(x_{<t}) \;=\; \sum_{v\in\mathcal{V}} P_\theta(v \mid x_{<t})\,G_\eta(x_{<t}v) \;=\; \sum_{v\in\mathcal{V}_t} P_\theta(v \mid x_{<t})\,G_\eta(x_{<t}v)$$

(terms with $G_\eta(x_{<t}v) = 0$ vanish), we obtain:

$$P_{\text{tilt+mask}}(x_t \mid x_{<t}) \;=\; P_\theta(x_t \mid x_{<t})\,\frac{G_\eta(x_{<t}x_t)}{G_\eta(x_{<t})} \;=\; Q^*(x_t \mid x_{<t}).$$

Equality of next-token conditionals at every prefix with positive probability under Q^* implies, by induction on t, equality of the induced sequence distributions: $\tilde{Q}_\phi = Q^*$.

Discussion. The result ties STaMP to the information-theoretic optimum: if the Soft Tilt supplies the ideal log-bias and the Mask implements exact P_θ-reachable viability (including EOS gating and length budgets), then the controlled decoder reproduces the I-projection exactly. In practice, the only approximation gaps arise from (i) deviations of b_ϕ from b^* and (ii) any relaxation in viability testing; when these vanish, so does the gap to Q^*.

A.7 No Dead-Ends Under Feasibility (Progress/Liveness)

Setting. Let $\mathcal{C} = \{x \in \mathcal{V}^* :\ \mathcal{R}\text{ accepts SVG}(x)\}$ be the hard feasible set recognized by $\mathcal{R}$ with accepting states F. As in equation 10, at step t the Mask restricts next tokens to the viable set:

$$\mathcal{V}_t \;=\; \Big\{\, v \in \mathcal{V} :\ \exists\, y \in \mathcal{V}^* \text{ such that } \delta\big(\delta(s_t, v), y\big) \in F \text{ and the termination rule is satisfied} \Big\}, \tag{17}$$

where s_t is the state of $\mathcal{R}$ after consuming $\text{SVG}(x_{<t})$, δ is the transition function composed with the token decoder, and the "termination rule" is either EOS-terminated (EOS is permitted iff $s_t \in F$) or fixed-length with remaining budget. The controlled one-step policy normalizes $P_{\text{tilt}}(\cdot \mid x_{<t}) \propto P_\theta(\cdot \mid x_{<t})\exp(b_\phi)$ over $\mathcal{V}_t$ (cf. equation 9–equation 10); we assume b_ϕ is finite so $\exp(b_\phi) > 0$. Throughout we assume feasibility from the main text: $P_\theta(\mathcal{C}) > 0$. We also adopt the standard softmax property of the base LM (cf. equation 2): for any prefix that occurs, $P_\theta(v \mid x_{<t}) > 0$ for all $v \in \mathcal{V}$.[1]

[1] If a model imposes structural zeros at the vocabulary level, replace $\mathcal{V}_t$ by $\{v \in \mathcal{V}_t :\ P_\theta(v \mid x_{<t}) > 0\}$; the argument below proceeds identically.

Claim (liveness). Under exact viability equation 17 with EOS gating (or length budgeting) and the feasibility assumption $P_\theta(\mathcal{C}) > 0$, the controlled decoder cannot stall. More precisely, along any trajectory generated by sampling from the Masked policy, the viable set is nonempty at every step before termination:

$$\mathcal{V}_t \neq \varnothing \quad \text{for all } t \text{ prior to halting,}$$

and in the EOS-terminated setting halting occurs only when $s_t \in F$ (so EOS is viable). Consequently the normalized policy $P_{\text{tilt+mask}}(\cdot \mid x_{<t})$ is well defined at every step until termination.

Proof. We argue by induction on t.

Base case. Feasibility provides some $x^\star \in \mathcal{C}$ with $P_\theta(x^\star) > 0$. In the EOS-terminated setting, this implies that from the start state s_1 there exists a valid continuation (namely $x^\star$), so by equation 17 the first symbol $v = x_1^\star$ is in $\mathcal{V}_1$; hence $\mathcal{V}_1 \neq \varnothing$. In the fixed-length setting, feasibility is interpreted with respect to the target length budget, yielding the same conclusion.

Inductive step. Suppose $\mathcal{V}_t \neq \varnothing$ and a token $x_t \in \mathcal{V}_t$ is sampled by the controller. Let $s_{t+1} = \delta(s_t, x_t)$. By the definition of $\mathcal{V}_t$, there exists a suffix y such that $\delta(s_{t+1}, y) \in F$ and the termination rule is satisfied for the remaining budget. If the process halts at t (EOS-terminated case), EOS must have been viable, which by gating implies $s_t \in F$; thus termination happens only at acceptance. If the process does not halt at t, we consider two cases:

(a) If $s_{t+1} \in F$, then EOS is permitted at $t{+}1$ by the gating rule; hence $\mathcal{V}_{t+1} \neq \varnothing$.

(b) If $s_{t+1} \notin F$, then any accepting completion y must be nonempty; let y_1 be its first token. By equation 17, $y_1 \in \mathcal{V}_{t+1}$, so $\mathcal{V}_{t+1} \neq \varnothing$.

This completes the induction.

Well-defined normalization. Because the base LM uses a softmax head, $P_\theta(v \mid x_{<t}) > 0$ for all $v \in \mathcal{V}$, and $\exp(b_\phi) > 0$ by assumption. Since $\mathcal{V}_t \neq \varnothing$ at each nonterminal step, the normalizing denominator $\sum_{v \in \mathcal{V}_t} P_{\text{tilt}}(v \mid x_{<t})$ is strictly positive, and $P_{\text{tilt+mask}}(\cdot \mid x_{<t})$ is a proper distribution.

Discussion. The result formalizes progress: exact viability ensures that from every nonterminal prefix on a trajectory, at least one next token keeps some accepting completion reachable, so the Mask never exhausts all options. The softmax property guarantees a positive normalizer, hence a well-defined sampling step. The argument is independent of the values of the Soft Tilt b_ϕ (in other words, any finite Soft Tilt cannot break liveness, it only re-weights viable options) and of any additional sampling heuristic, provided such heuristics are applied after masking and do not eliminate all viable tokens; thus liveness is fundamentally a property of the recognizer and the viability test coupled to the termination rule.

A.8 Efficiency comparison - STaMP vs. rejection / i.i.d. sampling

We compare expected computational cost measured in *token-generation steps* required to produce one valid (feasible) sequence in three regimes: STaMP (Soft Tilt + Mask), rejection sampling from P_θ, and any method that draws full sequences i.i.d. from P_θ and accepts only those in the feasible set C. We assume EOS-terminated generation and finite maximum length T (or, an almost-sure bound on length). The statements below extend to random yet integrable lengths by replacing T with the expected length.

Assumption. Let $C \subset V^*$ be the feasible set recognized by the deterministic recognizer R. Suppose: (*i*) $P_\theta(C) =: \varepsilon \in (0, 1)$ (the feasible set has base mass ε), (*ii*) STaMP enforces exact viability - every sample drawn from the masked controlled decoder $\widetilde{Q}$ lies in C with probability 1 (soundness), and the decoder always halts in at most T token steps (EOS-termination and bounded length), (*iii*) The cost of generating one token from the base model (including mask check) is counted as one token step, computing the mask is assumed to cost at most $O(1)$ token-step-equivalent work per token (so mask overhead is absorbed into the token-step count).

Claim (Rejection sampling is costly for rare feasible sets). Under the above assumption, the expected number of token-generation steps required by rejection sampling from P_θ to obtain one valid sequence is:

$$\mathbb{E}[\text{token steps}_{\text{rej}}] \;=\; \frac{T}{\varepsilon}.$$

Hence, when $\varepsilon \ll 1$ (rare feasible set) rejection sampling is inefficient: the expected token cost scales as $\frac{1}{\varepsilon}$.

Proof. Rejection sampling draws full sequences $X^{(1)}, X^{(2)}, \ldots$ i.i.d. from P_θ until one falls in C. Each draw produces a full sequence of at most T tokens (by assumption). The number of draws until the first success is geometric $\text{Geom}(\varepsilon)$

with mean $\frac{1}{\varepsilon}$. Therefore, the expected total token steps is the number of draws times T:

$$\mathbb{E}[\text{token steps}_{\text{rej}}] = T \cdot \mathbb{E}[\#\text{draws}] = T \cdot \frac{1}{\varepsilon}.$$

This gives the claimed result.

Claim (Any i.i.d. full-sequence sampler must account for $\frac{1}{\varepsilon}$ draws). Let a procedure produce candidate full sequences by drawing i.i.d. from P_θ and accept only those in C (this includes naive importance resampling that samples from P_θ then keeps only accepted draws). Then, the expected number of full-sequence draws until the first accepted sample is at least $\frac{1}{\varepsilon}$. Consequently, the expected token-step cost is at least $\frac{T}{\varepsilon}$.

Proof. Let N be the number of independent P_θ-draws needed to see the first sample in C. The probability of success on each draw is exactly ε, hence $N \sim \text{Geom}(\varepsilon)$ and $\mathbb{E}[N] = \frac{1}{\varepsilon}$. The corresponding token-step cost is $\mathbb{E}[N] \cdot T = \frac{T}{\varepsilon}$.

Comparison to STaMP. Under the above assumption, STaMP produces one valid sequence with at most T token steps (since it constructs a feasible sequence in a single run). Therefore, the token-step cost for STaMP is bounded by T. Combining Claims 1 and 2, when $\varepsilon \ll 1$ the expected token-step cost of STaMP is smaller than that of rejection/i.i.d. sampling by a factor on the order of $\frac{1}{\varepsilon}$. That is, when the feasible set is rare under the base model, STaMP avoids the $\frac{1}{\varepsilon}$ blow-up inherent to sequence-level i.i.d. sampling.

B Additional Qualitative Results

Figure 4: Inference-time control of StarVector `img2svg` model.

C End-to-End Design Generation Results

User studies have shown that designers care for three classes of constraints: (i) global layout, (ii) asset placement, and (iii) brand styling, which bundles colour palette, typeface choice, font colour and weight into a single visual identity knob. We therefore evaluate our model and approach on a combination of these constraints and also include unconstrained generation:

C.1 UNCONSTRAINED DESIGN GENERATION

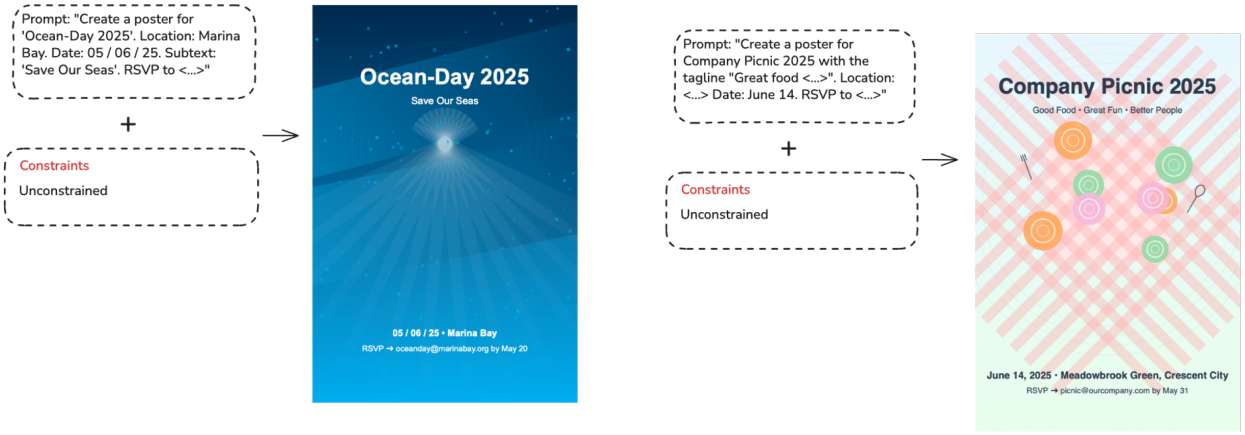

Figure 5: Unconstrained generation results (1/2).

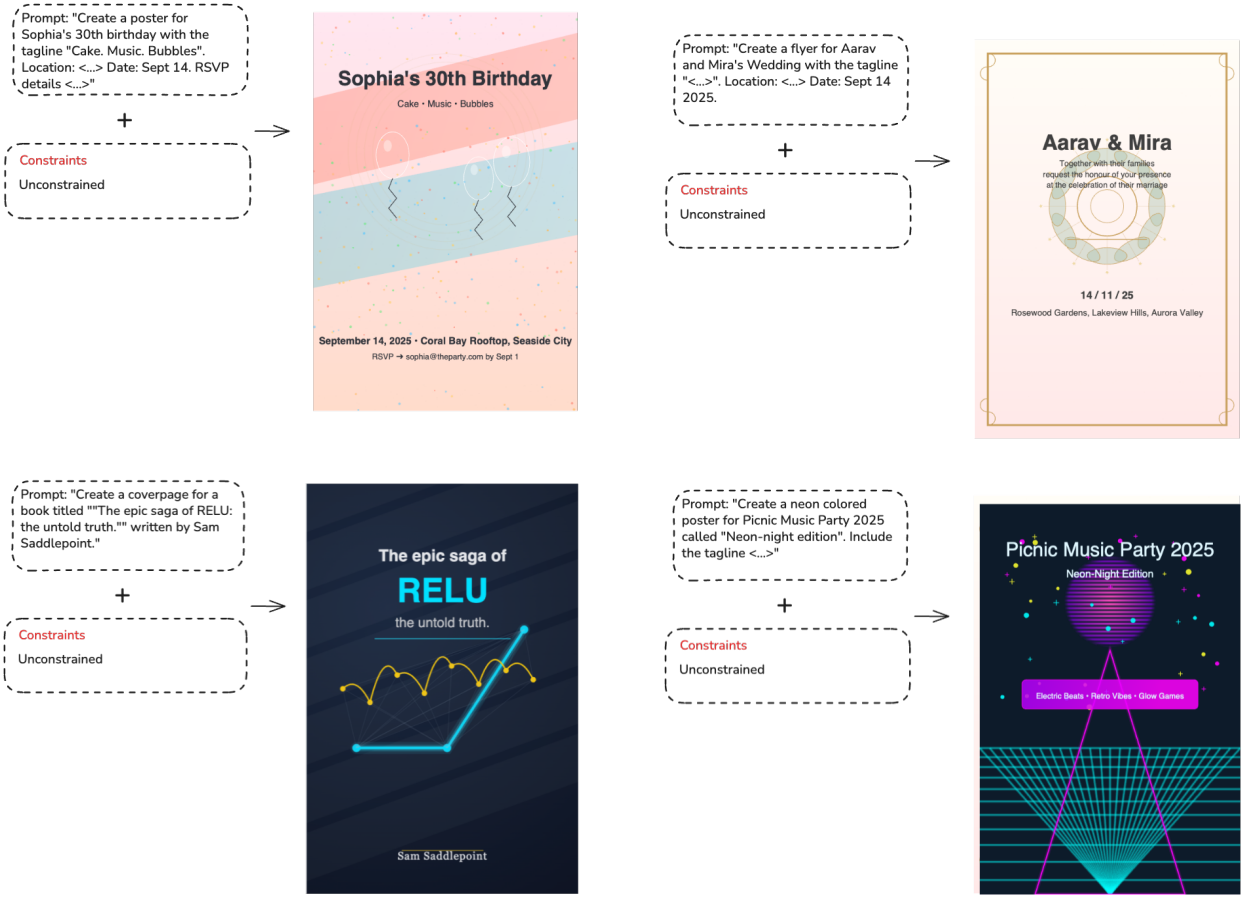

Figure 6: Unconstrained generation results.

C.2 CONSTRAINED DESIGN GENERATION

We now show the outputs on constrained generation.

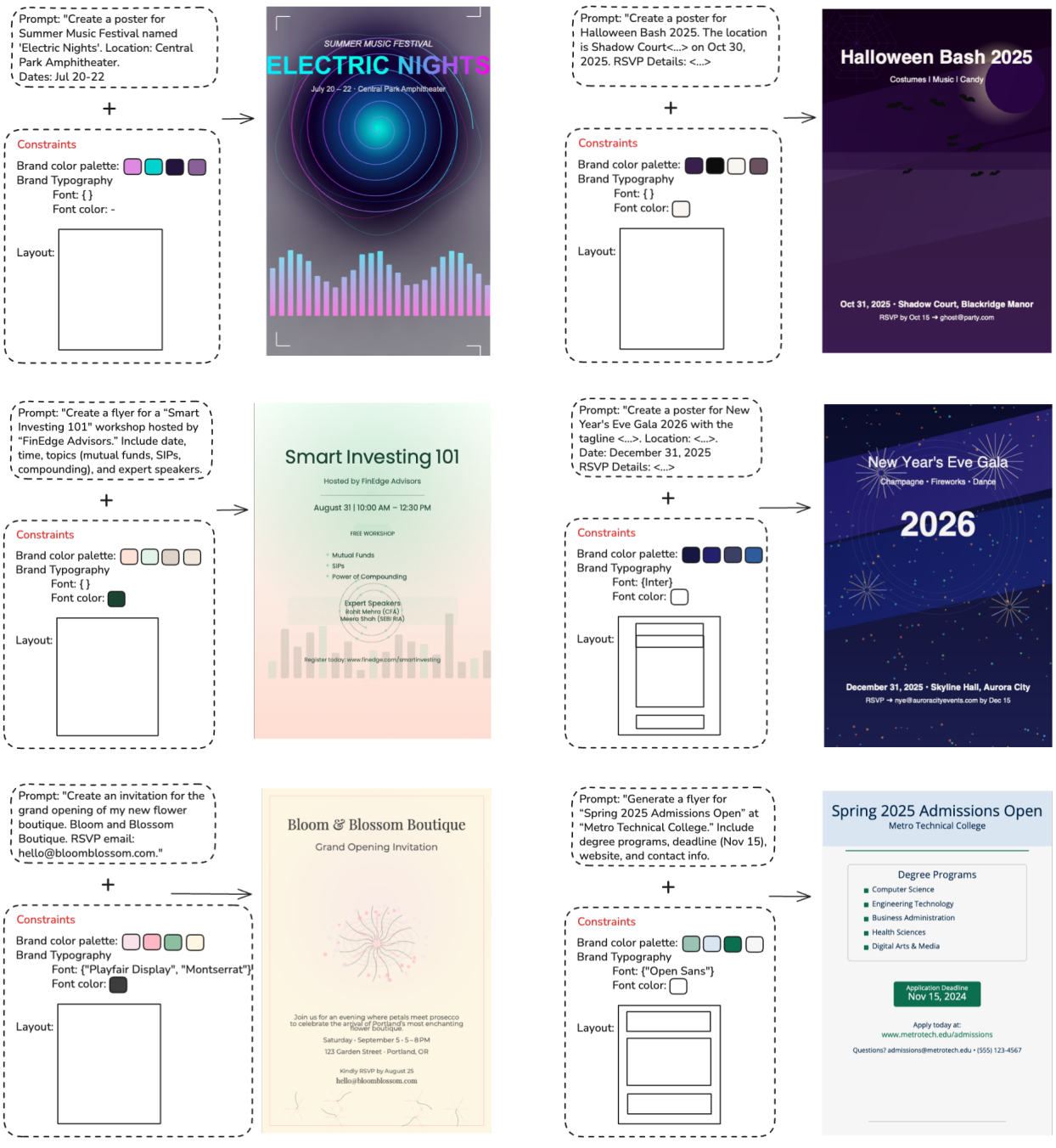

Figure 7: Constrained generation results (1/2).

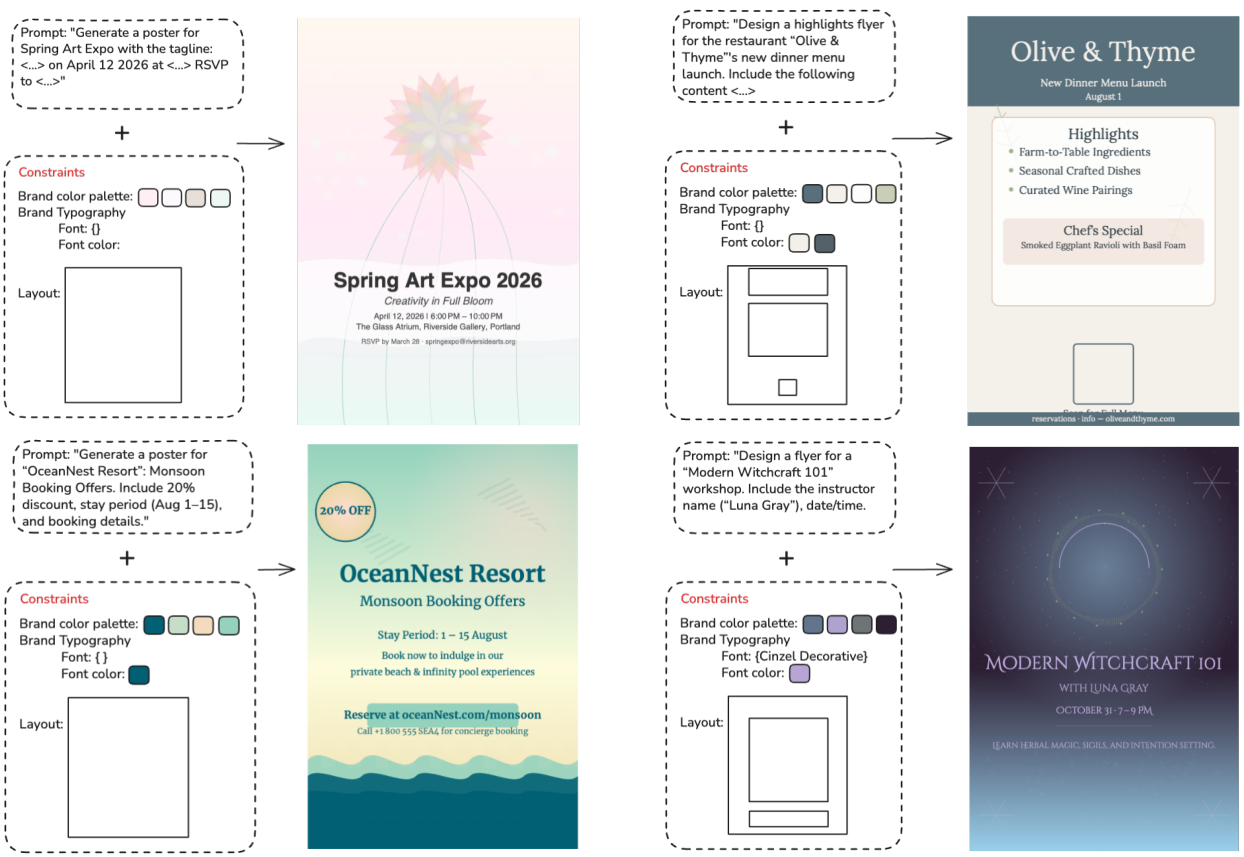

Figure 8: Constrained generation results (2/2).

D Design- and Control-Specific Evaluation Metrics

CIEDE 2000 (palette control). Let the target palette be $P^\star = \{c_1, \ldots, c_K\}$ (in CIE $L^*a^*b^*$) and the realized palette $\widehat{P} = \{\hat{c}_1, \ldots, \hat{c}_K\}$ extracted from SVG `fill`/`stroke` tokens (deduped by hex/RGB and mapped to $L^*a^*b^*$). Define the assignment

$$\pi^\star \;=\; \arg\min_{\pi \in S_K} \; \frac{1}{K} \sum_{i=1}^{K} \Delta E_{00}\big(c_i, \hat{c}_{\pi(i)}\big),$$

and report

$$\text{CIEDE2000} \;=\; \frac{1}{K} \sum_{i=1}^{K} \Delta E_{00}\big(c_i, \hat{c}_{\pi^\star(i)}\big),$$

where ΔE_{00} is the standard CIEDE 2000 color-difference in $L^*a^*b^*$ space (lower is better).

Layout Overlap % (region IoU). Let the specification provide N target regions with binary masks $\{M_i\}_{i=1}^N$ (in render pixel space), and let $\{\widehat{M}_i\}_{i=1}^N$ be realized masks obtained from the rendered SVG (via ID/color tags or raster segmentation). Define

$$\text{IoU}(M_i, \widehat{M}_i) \;=\; \frac{|M_i \cap \widehat{M}_i|}{|M_i \cup \widehat{M}_i|}, \qquad \text{Overlap\%} \;=\; 100 \cdot \frac{1}{N} \sum_{i=1}^{N} \text{IoU}(M_i, \widehat{M}_i).$$

Font Match % (typography control). For each specified text slot t with target font attributes $\text{font}_t^\star = (\texttt{family}, \texttt{weight}, \texttt{style})$, parse realized `<text>` nodes to recover $\widehat{\text{font}}_u$. Greedily match slots to nodes by maximum string overlap of textual content, then compute

$$\text{Font Match\%} \;=\; 100 \cdot \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}\{\widehat{\text{font}}_{u(t)} = \text{font}_t^\star\},$$

(optionally report a relaxed variant counting family-only matches).

CLIP Score (prompt alignment). Render the SVG to an image $R(x)$; compute text and image embeddings $e_t = \mathrm{CLIP}_{\text{text}}(p)$, $e_i = \mathrm{CLIP}_{\text{img}}(R(x))$. Report cosine similarity

$$\text{CLIP} \;=\; \frac{e_t^\top e_i}{\|e_t\|\,\|e_i\|}.$$

CLIP-Aesthetic (design quality proxy). Apply the LAION aesthetic predictor $a(\cdot)$ on the image embedding of $R(x)$:

$$\text{CLIP-Aesthetic} \;=\; a\big(\mathrm{CLIP}_{\text{img}}(R(x))\big),$$

(higher is better; model outputs are typically on a 1–10 scale).

ROUGE-L Recall (RLR) on SVG copy. Extract concatenated text $y(x)$ from all `<text>` nodes (reading order heuristic). Let p be the prompt string. With $\mathrm{LCS}(y, p)$ the longest common subsequence length,

$$\text{RLR} \;=\; \frac{\mathrm{LCS}(y(x),\, p)}{|p|}\,.$$

Quality retention (LPIPS). Compare the controlled rendering $R(\tilde{x})$ to the same backbone's uncontrolled rendering $R(x_{\text{base}})$ under identical prompts:

$$\text{LPIPS} \;=\; \mathrm{LPIPS}\big(R(\tilde{x}),\, R(x_{\text{base}})\big) \quad \text{(lower is better)},$$

and optionally a normalized retention score Retain $= 1 -$ LPIPS.

Token complexity. Sequence length in tokens:

$$\text{Tokens} \;=\; |x| \quad \text{and} \quad \Delta\text{Tokens} \;=\; |x| - |x_{\text{base}}|.$$

Generation time. Wall-clock latency per sample (seconds) measured end-to-end.

Well-Formedness Satisfaction Rate (WFSR). With a strict SVG parser $\mathcal{R}$,

$$\text{WFSR} \;=\; 100 \cdot \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}\{\mathcal{R} \text{ accepts } \mathrm{SVG}(x^{(n)})\}.$$

E USE OF LARGE LANGUAGE MODELS (LLMS).

We did not use LLMs for research ideation, algorithm design, or writing. Their role was limited to serving as an evaluation tool: we used LLMs to assess and compare outputs of our approach against baselines under specific design constraints, ensuring consistency and scalability in the evaluation process.