BCT: BAYES-CONSISTENT TEMPERING FOR FLOW LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Temperature scaling is typically applied as a post-hoc transformation of model probabilities. In posterior-induced flow language models, this creates two mismatches. First, posterior tempering implicitly changes both the endpoint distribution and the distribution of intermediate noisy states, while the denoiser remains trained on the original states. Second, the tempered posterior weights are combined with transport dynamics derived for the untempered flow. These mismatches are reflected empirically: lowering temperature improves GSM8K performance much less reliably in flow LMs than in discrete diffusion LMs. We introduce **Bayes-Consistent Tempering (BCT)**, which makes temperature explicit in both the distributions and dynamics of the generative flow. Its learned realization trains the denoiser on tempered intermediate states and tempers the endpoint distribution using the corresponding tempered dynamics. We also derive training-free variants for conventionally trained checkpoints. Across hyperspherical flow LMs with two denoiser architectures and a von Mises–Fisher flow LM, BCT improves GSM8K generation across broad temperature ranges and achieves the strongest tempering performance. The gains also extend to Sudoku, demonstrating benefits on tightly constrained generation. At very low temperatures, the learned realization is also markedly more robust than the training-free variants, supporting temperature as a property of the generative flow across training and inference rather than a post-hoc probability adjustment.

1 INTRODUCTION

Temperature scaling is one of the simplest controls for language generation (Ackley et al., 1985; Holtzman et al., 2020). In autoregressive models, dividing next-token logits by $T < 1$ directly sharpens the distribution from which the next token is sampled. In many flow language models for discrete data, however, generation is driven by a denoising posterior: a model predicts the clean endpoint token from an intermediate noisy state, and this posterior induces the velocity that transports the state toward token embeddings (Stark et al., 2024; Lee et al., 2026; Deschenaux & Gulcehre, 2026). Applying temperature only to this posterior creates two mismatches. First, by Bayes' rule, posterior tempering implicitly changes not only the endpoint distribution but also the *conditional bridge*—the time-indexed distribution of intermediate noisy states given each endpoint. The denoiser, however, is trained on states from the untempered bridge, so tempering changes the states encountered at inference without a corresponding change in training. Second, tempering changes the posterior weights over endpoints but leaves their conditional velocities unchanged, even though those velocities are designed for the original, untempered bridges. Thus, temperature changes the implied generative distribution without consistently changing how the model is trained or sampled.

This issue is also visible empirically. On GSM8K (Cobbe et al., 2021), prior work reports that flow language models gain little from lowering the temperature, in contrast to discrete diffusion models such as MDLM (Sahoo et al., 2024) and Duo (Sahoo et al., 2025), whose accuracy improves substantially under low-temperature decoding (Deschenaux & Gulcehre, 2026). This discrepancy motivates us to ask whether temperature is being incorporated correctly into posterior-induced flows.

We introduce **Bayes-Consistent Tempering (BCT)**, which treats temperature as part of the generative flow rather than as a post-hoc transformation of the denoising posterior. Its learned realization trains the denoiser on states from a tempered conditional bridge and, at inference, applies the

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

Bayes correction needed to temper the endpoint distribution while preserving that tempered bridge, together with the dynamics induced by the same bridge. We call the training component **Tempered-Flow Training (TFT)** and the corresponding sampler **BCT-L**.

We further ask how much of this consistency can be recovered without additional training after a model has been trained without tempering. Accepting the mismatch between intermediate states seen during training and inference, **BCT-P** retains post-hoc posterior tempering, which jointly tempers the endpoint distribution and bridge, while, as in BCT-L, matching the sampling dynamics to the tempered bridge. **BCT-E** takes a complementary approach: it keeps the original bridge and sampling dynamics, and tempers only the endpoint distribution. These training-free variants make BCT applicable to existing checkpoints while contrasting two ways to restore inference-time consistency: tempering the bridge with matched dynamics, or leaving the bridge unchanged.

For posterior-induced flow language models, BCT is not merely a conceptual framework: applying it requires deriving the tempered bridge and its corresponding sampling dynamics, and these quantities are tractable in concrete models. For $\mathbb{S}$-FLM (Deschenaux & Gulcehre, 2026), for example, the hyperspherical geometry lets us derive both explicitly. The same principle applies to SFCD (Chemseddine et al., 2026), which uses a von Mises–Fisher bridge with its own analytic tempering rule and corresponding dynamics. Thus, BCT is not specific to a particular denoiser architecture; its construction is determined by the bridge that defines the flow.

Compared with standard posterior tempering, BCT makes low-temperature decoding substantially more effective. Across hyperspherical flow LMs with two different denoiser architectures (S-arch/DiT) $\mathbb{S}$-FLM and the von Mises–Fisher flow LM (Chemseddine et al., 2026), BCT improves GSM8K generation over broad temperature ranges and achieves the strongest tempering results. The learned realization (BCT-L) is particularly robust under aggressive low-temperature sampling, where the training-free BCT variants deteriorate, showing the benefit of incorporating temperature into training rather than treating it only as an inference-time adjustment. The gains also extend to Sudoku (Ben-Hamu et al., 2025; Kim et al., 2026), where standard tempering does not improve at lower temperatures, suggesting that BCT's benefits extend to tightly constrained generation and are not specific to GSM8K-style reasoning.

2 Preliminaries

2.1 Posterior-induced flows

We consider a single token position with finite vocabulary $\mathcal{V}$ and endpoint distribution $\pi(x)$. Let $\boldsymbol{z}$ denote the continuous latent state and $t \in [0, 1]$ run from noise to data. For each endpoint token x, a **conditional bridge** $q_t(\boldsymbol{z} \mid x)$ describes the distribution of intermediate states at time t, starting from a token-independent source p_0 and converging toward its embedding $\boldsymbol{e}_x$. All bridge densities are defined with respect to a fixed reference measure μ on the latent space. A corresponding **conditional velocity** $u_t(\boldsymbol{z} \mid x)$ transports states so that their distribution follows $q_t(\cdot \mid x)$ over time; formally, q_t and u_t satisfy the continuity equation.

Bayes' rule gives the denoising posterior

$$r_t(x \mid \boldsymbol{z}) = \frac{\pi(x)\, q_t(\boldsymbol{z} \mid x)}{\sum_{y \in \mathcal{V}} \pi(y)\, q_t(\boldsymbol{z} \mid y)}, \tag{1}$$

and the generative velocity is its posterior-weighted average of the conditional velocities,

$$u_t(\boldsymbol{z}) = \sum_{x \in \mathcal{V}} r_t(x \mid \boldsymbol{z})\, u_t(\boldsymbol{z} \mid x). \tag{2}$$

A denoiser $r_t^\eta(x \mid \boldsymbol{z})$ is trained to approximate r_t and replaces it in Eq. (2) at generation time. We call flows constructed in this way **posterior-induced flows**.

2.2 Standard tempering

Writing $\beta = 1/T$, standard temperature scaling replaces the denoising posterior by

$$r_t^T(x \mid \boldsymbol{z}) \propto r_t(x \mid \boldsymbol{z})^\beta, \qquad u_t^T(\boldsymbol{z}) = \sum_x r_t^T(x \mid \boldsymbol{z})\, u_t(\boldsymbol{z} \mid x), \tag{3}$$

thus changing the posterior weights while leaving the conditional velocities unchanged. In practice, this is implemented by dividing the denoiser logits by T.

Using Eq. (1),

$$r_t(x \mid \boldsymbol{z})^\beta \propto \pi(x)^\beta q_t(\boldsymbol{z} \mid x)^\beta, \tag{4}$$

so posterior tempering acts algebraically on both the endpoint factor and the bridge likelihood. Define

$$\pi_T(x) \propto \pi(x)^\beta, \qquad q_{t,T}(\boldsymbol{z} \mid x) = \frac{q_t(\boldsymbol{z} \mid x)^\beta}{Z_{t,T}(x)}, \qquad Z_{t,T}(x) = \int q_t(\boldsymbol{z} \mid x)^\beta \, d\mu(\boldsymbol{z}). \tag{5}$$

The posterior induced by $(\pi_T, q_{t,T})$ differs from standard posterior tempering only by the token-dependent normalizer $Z_{t,T}(x)$. Hence the two coincide whenever $Z_{t,T}(x)$ is independent of x, as for the symmetric bridge families studied in this work.

For clarity, we present all derivations for a single token position. For sequences, the same decomposition applies position-wise, with $\pi(x)$ replaced by the context-conditioned endpoint distribution induced by the remaining positions; see Appendix A.

3 Bayes-Consistent Tempering

Once an endpoint distribution and a conditional bridge are specified, the posterior weights and sampling dynamics should correspond to that same generative model. In particular, for the tempered pair $(\pi_T, q_{t,T})$, the sampler should use its denoising posterior, conditional velocities $u_{t,T}(\boldsymbol{z} \mid x)$ that transport $q_{t,T}$, and the source $q_{0,T}$ of that bridge. However, standard tempering does not generally satisfy this alignment: while its posterior corresponds to the tempered pair under the condition described in Section 2, it continues to use the conditional velocities $u_t(\boldsymbol{z} \mid x)$ of the original bridge.

In contrast, our proposal, **Bayes-Consistent Tempering (BCT)**, treats the endpoint distribution, conditional bridge, and sampling dynamics as components of the same generative flow. We first develop its learned realization, which also aligns the distribution of states seen during training (Sec. 3.1), and then derive post-hoc variants for conventionally trained models (Sec. 3.2).

3.1 Learned BCT

The learned realization of BCT, **BCT-L**, aligns training with the tempered bridge used at inference. We first train the denoiser on intermediate states drawn from $q_{t,T}$, rather than from the original bridge q_t. Given a training distribution $p(T)$ over temperatures, **Tempered-Flow Training (TFT)** minimizes the usual denoising loss,

$$\mathcal{L}(\eta) = \mathbb{E}_{T\sim p(T),\, t,\, x\sim\pi,\, \boldsymbol{z}\sim q_{t,T}(\cdot\mid x)} \left[-\log r^\eta_{t,T}(x \mid \boldsymbol{z}) \right], \tag{6}$$

where the denoiser is conditioned on T. Since the endpoint token is still drawn from π while its intermediate state is drawn from tempered bridge $q_{t,T}$, the Bayes-optimal denoiser is

$$r_{t,T}(x \mid \boldsymbol{z}) = \frac{\pi(x)\, q_{t,T}(\boldsymbol{z} \mid x)}{\sum_y \pi(y)\, q_{t,T}(\boldsymbol{z} \mid y)}. \tag{7}$$

Thus, TFT incorporates the tempered bridge into the learned posterior, but does not itself temper the endpoint distribution.

At inference, BCT targets the pair $(\pi_T, q_{t,T})$. Its tempered posterior can be written directly in terms of the TFT posterior as

$$r^{\text{BCT-L}}_{t,T}(x \mid \boldsymbol{z}) \propto r_{t,T}(x \mid \boldsymbol{z})^\beta q_{t,T}(\boldsymbol{z} \mid x)^{1-\beta}. \tag{8}$$

The second factor is the Bayes correction that tempers the endpoint while preserving the tempered bridge already represented by the denoiser. In practice, learned posteriors are approximate, so we allow the strength of this correction to be selected on held-out data:

$$\ell^{\text{BCT-L}}_x = \beta\, \ell^\eta_x + \lambda(1-\beta) \log q_{t,T}(\boldsymbol{z} \mid x), \qquad 0 \le \lambda \le 1, \tag{9}$$

where one λ is selected per trained model and shared across temperatures.

Finally, BCT-L uses the tempered conditional velocity $u_{t,T}(\boldsymbol{z} \mid x)$ that transports the corresponding tempered bridge $q_{t,T}$:

$$u_{t,T}^{\text{BCT-L}}(\boldsymbol{z}) = \sum_x r_{t,T}^{\text{BCT-L}}(x \mid \boldsymbol{z})\, u_{t,T}(\boldsymbol{z} \mid x), \tag{10}$$

with sampling initialized from the tempered source $q_{0,T}$. The form of $u_{t,T}$ is model-dependent; we derive it for the flow constructions studied here in Sec. 3.3. Thus, TFT and BCT-L align the intermediate states seen during training, the endpoint distribution, and the sampling dynamics within the same generative flow.

3.2 Post-hoc BCT Variants for Conventionally Trained Models

We next ask how much of BCT can be recovered after a model has already been trained on the original bridge q_t. We consider two complementary post-hoc variants—**BCT-P** and **BCT-E**.

Path tempering (BCT-P). BCT-P retains posterior tempering, which implicitly tempers both the endpoint distribution and the bridge, but replaces the original conditional dynamics with those of the tempered bridge. Its posterior is

$$r_{t,T}^{\text{BCT-P}}(x \mid \boldsymbol{z}) \propto \frac{r_t(x \mid \boldsymbol{z})^\beta}{Z_{t,T}(x)}, \tag{11}$$

and the corresponding velocity is

$$u_{t,T}^{\text{BCT-P}}(\boldsymbol{z}) = \sum_x r_{t,T}^{\text{BCT-P}}(x \mid \boldsymbol{z})\, u_{t,T}(\boldsymbol{z} \mid x). \tag{12}$$

When $Z_{t,T}(x)$ is independent of x, BCT-P uses exactly the same posterior weights as standard tempering; the difference is that its conditional velocities are matched to the tempered bridge. Unlike BCT-L, however, its denoiser was trained only on states from the original bridge q_t, leaving a train–inference mismatch in the intermediate states.

Endpoint tempering (BCT-E). BCT-E avoids tempering the bridge altogether. It keeps untempered bridge and conditional dynamics, and tempers only the endpoint distribution π. By Bayes' rule, its posterior can be written using the original denoiser as

$$r_{t,T}^{\text{BCT-E}}(x \mid \boldsymbol{z}) \propto r_t(x \mid \boldsymbol{z})^\beta q_t(\boldsymbol{z} \mid x)^{1-\beta}, \tag{13}$$

or, in logits,

$$\ell_x^{\text{E}} = \beta\, \ell_x + (1-\beta) \log q_t(\boldsymbol{z} \mid x). \tag{14}$$

These weights are combined with the original conditional velocities $u_t(\boldsymbol{z} \mid x)$.

Both variants require no retraining. BCT-P restores consistency by matching the sampling dynamics to the tempered bridge, whereas BCT-E restores consistency by keeping the bridge and sampling dynamics untempered.

3.3 Instantiations

We instantiate BCT on two posterior-induced flow constructions with different conditional bridges.

Hyperspherical flow. S-FLM (Deschenaux & Gulcehre, 2026) is based on hyperspherical flow, which places token embeddings and latents on the unit sphere and transports a uniform source toward each endpoint along geodesics. Let $\theta_x(\boldsymbol{z}) = \arccos\langle \boldsymbol{z}, \boldsymbol{e}_x \rangle$ and $a_t = 1 - \alpha_t$ denote the remaining fraction of the geodesic at time t. Its conditional bridge has density

$$q_t(\boldsymbol{z} \mid x) \propto \frac{\sin^{d-2}(\theta_x / a_t)}{\sin^{d-2}\theta_x}\, \mathbf{1}\{\theta_x < a_t \pi\}, \tag{15}$$

with respect to the spherical surface measure. This density gives the bridge likelihood needed by BCT-E and TFT. Rotational symmetry also makes $Z_{t,T}(x)$ independent of x, so BCT-P uses the same posterior weights as standard tempering.

The tempered bridge $q_{t,T}$ remains rotationally symmetric around $\boldsymbol{e}_x$. Its corresponding conditional velocity therefore follows the same geodesic direction as the original S-FLM velocity, with a temperature-dependent change in speed,

$$u_{t,T}(\boldsymbol{z} \mid x) = w_{t,T}(\theta_x)\, u_t(\boldsymbol{z} \mid x), \tag{16}$$

where $w_{t,T}$ is obtained from a one-dimensional continuity equation; we give its derivation in Appendix B.3. The same rotational symmetry reduces exact sampling from $q_{t,T}$ to a one-dimensional transformation of the polar angle, allowing TFT to draw training states directly from the tempered bridge (Appendix B.4).

von Mises–Fisher flow. Chemseddine et al. (2026) use a von Mises–Fisher conditional bridge,

$$q_t(\boldsymbol{z} \mid x) \propto \exp(\kappa_t \langle \boldsymbol{e}_x, \boldsymbol{z} \rangle)\,. \tag{17}$$

Tempering stays within the same family:

$$q_{t,T}(\boldsymbol{z} \mid x) \propto \exp\Big(\frac{\kappa_t}{T} \langle \boldsymbol{e}_x, \boldsymbol{z} \rangle\Big)\,, \tag{18}$$

so BCT simply replaces κ_t by κ_t/T and uses the corresponding conditional dynamics (Appendix G). TFT likewise draws training states directly from the vMF distribution with concentration κ_t/T. The normalizer is independent of x, and BCT-P therefore again shares its posterior weights with standard tempering.

For von Mises–Fisher flow, the tempered bridge evidence $g_{t,T}(x, \boldsymbol{z}) = (\kappa_t/T)\langle \boldsymbol{e}_x, \boldsymbol{z} \rangle$ is available analytically. For TFT, we therefore include this term explicitly in the denoiser logits, $\ell^{\eta}_x = a_{\eta,x} + g_{t,T}(x, \boldsymbol{z})$, and learn only the remaining component $a_{\eta,x}$. This is a parameterization of the same TFT posterior and BCT-L sampler, not a separate training method.

4 Experiments

4.1 Experimental Setup

Main setting. Our main experiments evaluate BCT on GSM8K (Cobbe et al., 2021) with $\mathbb{S}$-FLM (Deschenaux & Gulcehre, 2026). Following its setup, we adopt the TinyGSM protocol (Liu et al., 2023): for each of the 1319 test problems, the model generates a Python program, and the problem counts as solved if executing the program returns the reference answer. We use its released TinyGSM checkpoints for two denoiser backbones: S-arch, a Transformer whose intermediate activations lie on the hypersphere, and the standard Diffusion Transformer (DiT; Peebles & Xie, 2023). We refer to either checkpoint as $\mathbb{S}$-FLM and specify the backbone where relevant. The two backbones share the bridge, the posterior-induced velocity, and all samplers, so comparing them varies only the denoiser architecture.

Additional settings. Two additional experiments test whether our conclusions generalize to other settings (Sections 4.4 and 4.5). First, we change the task while keeping the hyperspherical flow, using Sudoku-Hard (Kim et al., 2026; Ben-Hamu et al., 2025), which requires tightly constrained generation. Since no Sudoku checkpoint is released, we train a DiT-based $\mathbb{S}$-FLM with its released training recipe (Appendix F). Second, we change the flow structure while keeping the task: we apply BCT to SFCD (Chemseddine et al., 2026), which uses a von Mises–Fisher bridge instead of the geodesic bridge of $\mathbb{S}$-FLM, and evaluate its released TinyGSM checkpoint on the same GSM8K problems (Appendix G). Unless noted otherwise in their sections, both follow the protocol below.

Samplers. All samplers use the exact posterior-induced velocity with 1024 function evaluations (NFE) and otherwise follow the released sampler; they differ only in the posterior weights and conditional velocities. We compare standard tempering Eq. (3), BCT-E Eq. (14), BCT-P Eq. (12), and BCT-L, which is described below. As a reference, we report the Top-1 velocity (Deschenaux & Gulcehre, 2026), which moves only toward the most likely token; it is invariant to temperature and is the $T \to 0$ limit of standard tempering. We evaluate $T \in \{1, 0.8, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$ in all experiments.

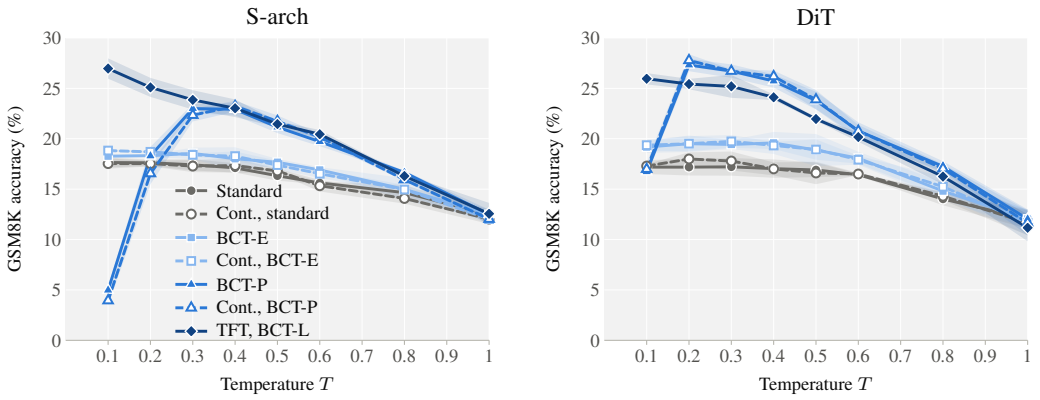

Figure 1: GSM8K accuracy against the sampling temperature on the two backbones (NFE $= 1024$). Lines are the mean over three generation seeds and bands $\pm$ one sample standard deviation. Curves without a prefix use the released checkpoint; Cont. (dashed) is the matched continuation. At $T = 1$ every BCT correction is the identity. Values in Table 1.

Table 1: GSM8K accuracy (%), mean $\pm$ sample standard deviation over three generation seeds (1319 problems, NFE $= 1024$). Columns with the same sampler (Cont. vs. $\mathbb{S}$-FLM) isolate the effect of continual training; columns with the same checkpoint isolate the effect of the sampler. At $T = 1$ every correction is the identity. Top-1 reference at seed 1: 17.2 (S-arch) and 16.4 (DiT).

	TFT	Cont.			$\mathbb{S}$-FLM		
T	BCT-L	Standard	BCT-E	BCT-P	Standard	BCT-E	BCT-P
				S-arch			
1	12.6 $\pm$1.1	12.0 $\pm$0.5	12.0 $\pm$0.5	12.0 $\pm$0.5	12.5 $\pm$0.5	12.5 $\pm$0.5	12.5 $\pm$0.5
0.8	16.3 $\pm$0.2	14.1 $\pm$0.5	15.0 $\pm$0.5	15.9 $\pm$0.2	14.7 $\pm$0.7	15.0 $\pm$1.1	16.7 $\pm$0.2
0.6	20.4 $\pm$0.4	15.3 $\pm$0.6	16.5 $\pm$0.4	20.2 $\pm$0.3	15.6 $\pm$0.2	16.9 $\pm$0.3	19.7 $\pm$0.5
0.5	21.5 $\pm$0.3	16.8 $\pm$0.2	17.4 $\pm$0.4	21.7 $\pm$0.6	16.3 $\pm$0.5	17.7 $\pm$0.3	21.2 $\pm$0.6
0.4	23.0 $\pm$0.9	17.3 $\pm$0.7	18.3 $\pm$0.5	23.2 $\pm$0.5	17.1 $\pm$0.5	18.0 $\pm$1.1	22.9 $\pm$0.6
0.3	23.9 $\pm$1.0	17.3 $\pm$0.6	18.4 $\pm$0.2	22.3 $\pm$0.6	17.4 $\pm$0.6	18.5 $\pm$0.6	23.0 $\pm$0.5
0.2	25.1 $\pm$0.9	17.5 $\pm$0.3	18.7 $\pm$0.1	16.6 $\pm$1.1	17.6 $\pm$0.3	18.3 $\pm$0.1	18.3 $\pm$0.9
0.1	27.0 $\pm$1.0	17.5 $\pm$0.5	18.8 $\pm$0.1	3.9 $\pm$0.5	17.6 $\pm$0.5	18.3 $\pm$0.1	5.0 $\pm$0.4
				DiT			
1	11.2 $\pm$1.3	11.7 $\pm$1.2	11.7 $\pm$1.2	11.7 $\pm$1.2	12.0 $\pm$1.0	12.0 $\pm$1.0	12.0 $\pm$1.0
0.8	16.2 $\pm$0.8	14.3 $\pm$0.4	15.2 $\pm$1.0	17.0 $\pm$0.4	14.0 $\pm$0.6	14.8 $\pm$0.9	17.3 $\pm$0.5
0.6	20.2 $\pm$0.5	16.5 $\pm$0.1	17.9 $\pm$0.4	20.8 $\pm$0.6	16.4 $\pm$0.1	18.1 $\pm$0.6	20.8 $\pm$0.9
0.5	22.0 $\pm$0.2	16.6 $\pm$1.1	18.9 $\pm$1.0	23.9 $\pm$0.9	16.9 $\pm$0.7	18.9 $\pm$1.6	23.8 $\pm$0.9
0.4	24.1 $\pm$0.3	17.0 $\pm$0.6	19.3 $\pm$0.3	26.2 $\pm$0.6	17.0 $\pm$0.9	19.6 $\pm$1.1	25.7 $\pm$0.7
0.3	25.2 $\pm$1.1	17.8 $\pm$0.9	19.7 $\pm$0.6	26.7 $\pm$0.2	17.2 $\pm$0.9	19.4 $\pm$0.5	26.7 $\pm$0.7
0.2	25.4 $\pm$0.5	18.0 $\pm$0.7	19.5 $\pm$0.7	27.8 $\pm$0.6	17.2 $\pm$0.8	19.5 $\pm$0.8	27.3 $\pm$0.6
0.1	26.0 $\pm$0.6	17.3 $\pm$0.2	19.4 $\pm$0.6	17.0 $\pm$0.6	17.2 $\pm$0.7	19.2 $\pm$0.6	16.7 $\pm$0.4

TFT and BCT-L. For each backbone, we continue training the released checkpoint with TFT Eq. (6) for 2,000 updates, 0.8% of its 250k pretraining updates, and refer to the resulting model as TFT. The temperature distribution $p(T)$ draws $T = 1$ with probability 0.2, $T = 0.1$ with probability 0.1, and otherwise $\beta \sim U(1.25, 5)$, so it covers every evaluated temperature. As a matched training control, Cont. continues training for the same 2,000 updates with the original recipe. All models are evaluated with EMA weights. λ in Eq. (9) is selected once per model on TinyGSM validation problems and shared across temperatures (Appendix C).

4.2 Main Results

Figure 1 and Table 1 show the results for the hyperspherical flow, whose bridge moves latents along geodesics on the sphere (Section 3.3), with both denoiser backbones. Since TFT continues a pre-

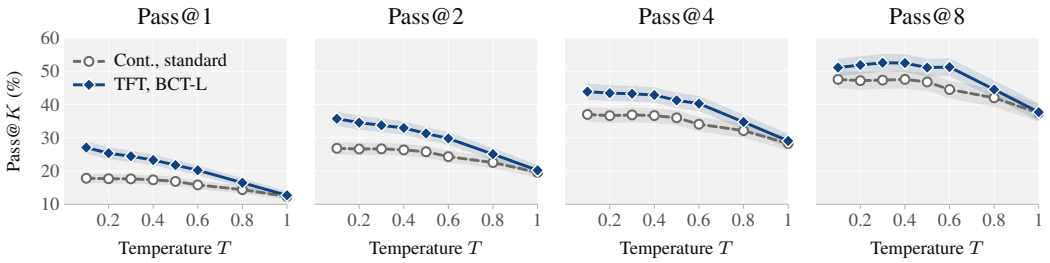

Figure 2: Pass@K on GSM8K (S-arch, NFE $= 1024$), estimated from eight generations per problem. Bands are 95% bootstrap intervals over problems; the panels share the vertical axis.

trained model rather than training from scratch, the fair baseline for BCT-L is Continual Training (Cont.), which receives the same additional update steps with the original recipe; the two differ only in whether temperature enters training. The continuation itself changes little: standard tempering on Cont. stays within one point of the released checkpoint at every temperature, and the training-free variants behave alike on the two checkpoints.

BCT-L makes low-temperature decoding effective. With conventional training, lowering the temperature helps little: standard tempering stays at 17–18% for every $T \leq 0.4$ on both backbones, close to the Top-1 reference, its $T \to 0$ limit. BCT-L instead improves steadily as the temperature decreases, reaching $27.0 \pm 1.0\%$ on S-arch and $26.0 \pm 0.6\%$ on DiT at $T = 0.1$, and it is more accurate than standard tempering at every $T < 1$.

Matching the dynamics at inference gives large gains at moderate temperatures. Without training, BCT-P already improves substantially on standard tempering, raising its best accuracy by about 6 points on S-arch and 10 points on DiT, and at moderate temperatures it is comparable to BCT-L, slightly ahead on DiT.

Strong tempering requires training on the tempered bridge. Under strong tempering, however, BCT-P collapses and falls below standard tempering at $T = 0.1$ on both backbones. This is consistent with its train–inference mismatch (Section 3.2): the latents follow the tempered bridge, on whose states the denoiser was not trained, and the mismatch grows as T decreases. BCT-L, whose denoiser is trained on the tempered bridge, has no such mismatch and is the most accurate sampler at $T = 0.1$. The results at $T = 0.1$ thus particularly highlight the importance of removing the mismatch between the intermediate states seen in training and those visited during sampling.

Keeping the bridge is stable but gains less. BCT-E keeps the bridge on which the denoiser was trained and therefore does not degrade at low temperature, but it improves only modestly on standard tempering and stays below BCT-L at every $T < 1$, increasingly so as the temperature decreases.

Consistency at inference thus helps in both forms: matching the bridge and sampling dynamics gives large gains at moderate temperatures, and keeping the bridge avoids the failure but gains little. Only BCT-L, which also brings the tempered bridge into training, is both accurate and robust under strong tempering, e.g., $T = 0.1$.

4.3 Gains Beyond Reduced Diversity

Lowering the temperature concentrates the sampler, so the low-temperature gain of BCT-L could merely trade sample diversity for single-sample accuracy (Chen et al., 2021), which a less concentrated sampler would recover with more samples. We test this on S-arch by comparing BCT-L with standard tempering on Cont. in Pass@K for $K \in \{1, 2, 4, 8\}$, estimated from eight generations per problem with the unbiased estimator of Chen et al. (2021) (Figure 2; details in Appendix D).

For every $T < 1$ and $K \leq 8$, BCT-L has the higher Pass@K, and the 95% paired bootstrap interval of the difference lies above zero. More samples raise Pass@K for both samplers, but not enough for standard tempering to catch up. The low-temperature gain of BCT-L is therefore not explained by reduced diversity alone.

4.4 BEYOND GSM8K: SUDOKU-HARD

Sudoku-Hard tests whether our conclusions carry over to tightly constrained generation, in which each puzzle has a single valid completion. Because no Sudoku checkpoint is released, we train the base model from scratch and obtain TFT by continuing it; as in Section 4.2, the fair baseline for BCT-L is therefore Cont. (Figure 3). Following the released configuration of $\mathbb{S}$-FLM (Deschenaux & Gulcehre, 2026), we sample with 180 NFE.

Lowering the temperature slightly hurts standard tempering, whereas BCT-L improves monotonically to 46.8%. The training-free variants also benefit from lower temperatures, and on this task BCT-P does not collapse; still, BCT-L is the most accurate at every $T \leq 0.6$, with a margin that widens as T decreases, and the base checkpoint shows the same pattern (Appendix F).

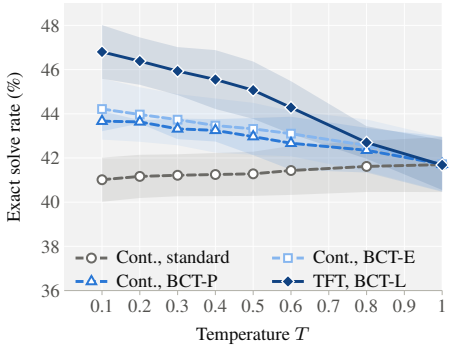

Figure 3: Sudoku-Hard exact solve rate (NFE $= 180$): mean and $\pm$ one sample standard deviation over three generation seeds; base-checkpoint results are in Table 5.

The conclusions of Section 4.2 thus hold for tightly constrained generation: BCT-L makes low temperatures effective where standard tempering does not.

4.5 BEYOND THE GEODESIC BRIDGE: VON MISES–FISHER FLOWS

We finally apply BCT to SFCD (Chemseddine et al., 2026), whose von Mises–Fisher bridge replaces the geodesic bridge of $\mathbb{S}$-FLM (Section 3.3). We use its released TinyGSM checkpoint and sampler with NFE $= 1024$, and updates for TFT and Cont., and average over five generation seeds (Appendix G). Unlike $\mathbb{S}$-FLM, this flow already benefits from standard tempering: lowering the temperature raises its accuracy by 9 points between $T = 1$ and $T = 0.3$ (Figure 4), suggesting that inconsistent tempering harms the vMF bridge less. We therefore ask whether consistent tempering still improves a model whose standard tempering is already effective.

It does, mainly at moderate temperatures. BCT-L improves on standard tempering at every $T \in [0.2, 0.6]$ and attains the highest accuracy among the evaluated samplers, $34.5 \pm 0.5\%$ at $T = 0.3$, against $32.6 \pm 0.6\%$ for standard tempering, a margin well beyond the seed standard deviations.

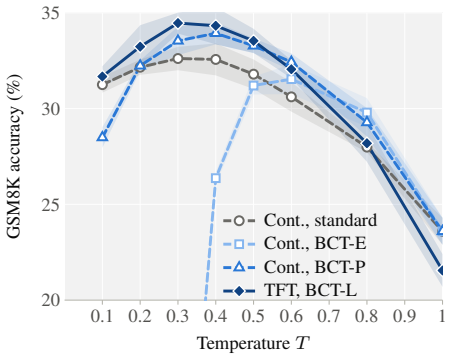

Figure 4: GSM8K accuracy of BCT on the vMF flow LM (NFE $= 1024$): mean and $\pm$ one sample standard deviation over five generation seeds. BCT-E falls below the plotted range for $T \leq 0.3$; more detailed results are in Table 7.

Under strong tempering, the learned realization remains stable while the training-free variants fail. BCT-L declines at the lowest temperature, a trend absent on $\mathbb{S}$-FLM, but only to the level of standard tempering (31.7% vs. 31.3% at $T = 0.1$). Both training-free variants instead fall below it: BCT-P as on $\mathbb{S}$-FLM (28.5%), and BCT-E, which is stable on $\mathbb{S}$-FLM, collapses to essentially zero accuracy for $T \leq 0.2$, likely because on this bridge its correction $-(\beta - 1)\kappa_t \langle \boldsymbol{e}_x, \boldsymbol{z} \rangle$ amplifies any mismatch between the learned logits and the analytic bridge term by β and κ_t. BCT-L is thus the only variant that never falls below standard tempering for $T < 1$: consistent tempering improves even a flow that already responds well to temperature, and training on the tempered bridge keeps it reliable under strong tempering.

5 Related Work

Posterior-induced flow language models. Non-autoregressive language models include discrete diffusion (Austin et al., 2021; Lou et al., 2024; Sahoo et al., 2024; Shi et al., 2024; Sahoo et al., 2025; Arriola et al., 2025; Nie et al., 2025) and continuous diffusion and flow models, which map tokens to continuous representations such as learned embeddings or one-hot vectors (Li et al., 2022; Dieleman et al., 2022; Gulrajani & Hashimoto, 2023; Lee et al., 2026; Hu et al., 2026). In many continuous models, a denoiser trained with cross-entropy predicts the clean token, and its posterior weights conditional velocities, drifts, or scores. Among spherical constructions (Davis et al., 2024; Jo & Hwang, 2025), $\mathbb{S}$-FLM moves latents along geodesics of the hypersphere (Deschenaux & Gulcehre, 2026; Chen & Lipman, 2024), whereas SFCD uses von Mises–Fisher bridges (Chemseddine et al., 2026); we instantiate BCT for both.

Temperature and consistency. In autoregressive models and discrete diffusion, temperature rescales the categorical distribution from which tokens are sampled (Ackley et al., 1985; Holtzman et al., 2020), and lowering it substantially improves discrete diffusion language models on GSM8K (Deschenaux & Gulcehre, 2026); we reproduce this in Appendix E. Continuous diffusion and flow models instead temper the score or the token posterior (Dieleman et al., 2022; Gulrajani & Hashimoto, 2023; Deschenaux & Gulcehre, 2026; Azangulov & Zhang, 2026), and for continuous data tempering is known not to commute with noising: the scaled score is not the score of the noised tempered distribution (Du et al., 2023). Work on this inconsistency targets other settings: Feynman–Kac correctors reweight and resample populations of particles along annealed marginals (Skreta et al., 2025; Hasan et al., 2026), Li et al. (2026) correct score scaling under Gaussian noising for flattening ($T > 1$), and temporal score rescaling controls a local temperature that leaves the mixture weights unchanged (Xu et al., 2026). None of them addresses flows induced by a token posterior over a finite vocabulary, where tempering the posterior implicitly changes the conditional bridge while leaving the conditional velocities untempered. BCT corrects this through the bridge itself, along a single sampling trajectory without particles, and extends the correction to training with TFT.

Train–inference mismatch. Models evaluated on states that differ from those seen in training suffer from exposure bias, both in autoregressive models (Ranzato et al., 2016; Bengio et al., 2015) and in diffusion models, where it is reduced by perturbing training inputs or shifting sampling time steps (Ning et al., 2023; Li et al., 2024). There, the mismatch arises from the model's own prediction errors. Under tempering, it is instead introduced by a known change of the bridge, so TFT can draw its training states exactly from the tempered bridge that the sampler follows.

6 Conclusion

We introduced **Bayes-Consistent Tempering (BCT)**, which treats temperature as part of the generative flow rather than as a post-hoc transformation of the denoising posterior. BCT aligns the tempered endpoint distribution, intermediate-state distribution, training distribution, and sampling dynamics; its training-free variants recover parts of this consistency for conventionally trained checkpoints. Across hyperspherical flow LMs with two denoiser architectures and the von Mises–Fisher flow LM, BCT improves GSM8K generation over broad temperature ranges, while its learned realization remains robust under aggressive tempering where the post-hoc variants deteriorate. The gains also extend to Sudoku, suggesting that BCT's benefits extend to tightly constrained generation. Overall, our results indicate that temperature in posterior-induced flow language models is better treated as part of the flow itself, across both training and inference.

Limitations and Discussion. BCT applies to posterior-induced flows for which the tempered bridge and its corresponding sampling dynamics can be derived or approximated; the concrete derivations in this work cover the hyperspherical bridge of $\mathbb{S}$-FLM and the von Mises–Fisher bridge of SFCD. Extending BCT to other bridge families is an important direction for future work.

Ethics Statement

This work proposes a temperature-scaling method for flow language models and evaluates it on publicly available data: GSM8K, TinyGSM, and Sudoku puzzles produced by a public generator. It involves no human subjects or personal data, and we are not aware of ethical concerns specific to this work.

Reproducibility Statement

Section 4.1 specifies the models, samplers, temperatures, and training budgets shared by all experiments. All GSM8K experiments start from publicly released checkpoints and use the released samplers and evaluation (Section 4.1, Appendix G); the Sudoku-Hard model is trained with the released recipe (Appendix F). Section 3.3 and Appendices B and G give the tempered bridges, conditional velocities, and exact tempered sampling needed to implement BCT, and Appendix C gives the held-out selection of λ. Appendices D–G list the evaluation protocols, seeds, and per-temperature results of every reported setting. We will release our code upon acceptance.

AI Use Statement

We used an LLM-based assistant to polish the writing of this paper. The research question, the method, and the derivations were developed by the authors, who take full responsibility for the content of this paper.

References

David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985.

Marianne Arriola, Aaron Gokaslan, Justin T Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, 2021.

Iskander Azangulov and Leo Zhang. Sampling from flow language models via marginal-conditioned bridges. *arXiv preprint arXiv:2605.13681*, 2026.

Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas Nolte, and Brian Karrer. Accelerated sampling from masked diffusion models via entropy bounded unmasking, 2025. URL https://arxiv.org/abs/2505.24857.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, 2015.

Jannis Chemseddine, Gregor Kornhardt, and Gabriele Steidl. Spherical flows for sampling categorical data. *arXiv preprint arXiv:2605.05629*, 2026.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *The Twelfth International Conference on Learning Representations*, 2024.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Oscar Davis, Samuel Kessler, Mircea Petrache, İsmail İlkan Ceylan, Michael Bronstein, and Avishek Joey Bose. Fisher flow matching for generative modeling over discrete data. In *Advances in Neural Information Processing Systems*, 2024.

Justin Deschenaux and Caglar Gulcehre. Language modeling with hyperspherical flows. *arXiv preprint arXiv:2605.11125*, 2026.

Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H. Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, Curtis Hawthorne, Rémi Leblond, Will Grathwohl, and Jonas Adler. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022.

Yilun Du, Conor Durkan, Robin Strudel, Joshua B. Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and MCMC. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8489–8510, 2023.

Ishaan Gulrajani and Tatsunori B. Hashimoto. Likelihood-based diffusion language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Mohsin Hasan, Viktor Ohanesian, Artem Gazizov, Yoshua Bengio, Alán Aspuru-Guzik, Roberto Bondesan, Marta Skreta, and Kirill Neklyudov. Discrete Feynman-Kac correctors. *arXiv preprint arXiv:2601.10403*, 2026.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020.

Keya Hu, Linlu Qiu, Yiyang Lu, Hanhong Zhao, Tianhong Li, Yoon Kim, Jacob Andreas, and Kaiming He. Elf: Embedded language flows. *arXiv preprint arXiv:2605.10938*, 2026.

Jaehyeong Jo and Sung Ju Hwang. Continuous diffusion model for language modeling. In *Advances in Neural Information Processing Systems*, 2025.

Jaeyeon Kim, Seunggeun Kim, Taekyun Lee, David Z. Pan, Hyeji Kim, Sham Kakade, and Sitan Chen. Fine-tuning masked diffusion for provable self-correction. In *International Conference on Machine Learning*, 2026.

Chanhyuk Lee, Jaehoon Yoo, Manan Agarwal, Sheel Shah, Jerry Huang, Aditi Raghunathan, Seunghoon Hong, Nicholas M Boffi, and Jinwoo Kim. Flow map language models: One-step language modeling via continuous denoising. *arXiv preprint arXiv:2602.16813*, 2026.

Mingxiao Li, Tingyu Qu, Ruicong Yao, Wei Sun, and Marie-Francine Moens. Alleviating exposure bias in diffusion models through sampling with shifted time steps. In *International Conference on Learning Representations*, 2024.

Peizhuo Li, Emre Aksan, Alexandru-Eugen Ichim, Thabo Beeler, and Olga Sorkine-Hornung. Diversify diffusion with temperature sampling and variance-corrective time shifting. *arXiv preprint arXiv:2607.10853*, 2026.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35: 4328–4343, 2022.

Bingbin Liu, Sebastien Bubeck, Ronen Eldan, Janardhan Kulkarni, Yuanzhi Li, Anh Nguyen, Rachel Ward, and Yi Zhang. TinyGSM: achieving >80% on GSM8K with small language models. *arXiv preprint arXiv:2312.09241*, 2023.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning*, 2024.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

Mang Ning, Enver Sangineto, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Input perturbation reduces exposure bias in diffusion models. In *International Conference on Machine Learning*, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. URL https://arxiv.org/abs/2212.09748.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations*, 2016.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin T Chiu, and Volodymyr Kuleshov. The diffusion duality. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=9P9Y8FOSOk.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

Marta Skreta, Tara Akhound-Sadegh, Viktor Ohanesian, Roberto Bondesan, Alán Aspuru-Guzik, Arnaud Doucet, Rob Brekelmans, Alexander Tong, and Kirill Neklyudov. Feynman-Kac correctors in diffusion: Annealing, guidance, and product of experts. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 55906–55949, 2025.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to dna sequence design. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 46495–46513, 2024.

Yanbo Xu, Yu Wu, Sungjae Park, Zhizhuo Zhou, and Shubham Tulsiani. Temporal score rescaling for temperature sampling in diffusion and flow models. In *International Conference on Machine Learning*, 2026.

A SEQUENCES

For a sequence $x = (x^1, \dots, x^L)$ with joint endpoint distribution $\pi(x)$ and a bridge that factorizes over positions, the posterior at position l marginalizes the other positions:

$$r_t(x^l \mid \boldsymbol{z}) = \frac{q_t(\boldsymbol{z}^l \mid x^l)\, \tilde{\pi}_t(x^l; \boldsymbol{z}^{-l})}{\sum_y q_t(\boldsymbol{z}^l \mid y)\, \tilde{\pi}_t(y; \boldsymbol{z}^{-l})}, \qquad \tilde{\pi}_t(x^l; \boldsymbol{z}^{-l}) = \sum_{x^{-l}} \pi(x) \prod_{j \neq l} q_t(\boldsymbol{z}^j \mid x^j). \tag{19}$$

This is Eq. (1) at position l with π replaced by the context-conditioned endpoint $\tilde{\pi}_t(\cdot\,; \boldsymbol{z}^{-l})$, which does not depend on $\boldsymbol{z}^l$. Every construction of Section 3 therefore applies position-wise and tempers $\tilde{\pi}_t$; under TFT, $\tilde{\pi}_t$ is formed with the tempered bridges $q_{t,T}$ at the other positions. None of them tempers the joint endpoint: that would require the context factor $\sum_{x^{-l}} \pi(x)^\beta \prod_{j \neq l} q'_t(\boldsymbol{z}^j \mid x^j)$, with q'_t the bridge of the target pair, which in general is not proportional to the power of $\tilde{\pi}$ that every variant uses. Per-token temperature in autoregressive models has the same property.

B THE HYPERSPHERICAL BRIDGE

Throughout, $\theta = \theta_x(\boldsymbol{z})$ is the polar angle about $\boldsymbol{e}_x$, $\nu = d - 2$, σ is the surface measure on $\mathbb{S}^{d-1}$, and $a = a_t \in (0, 1]$, which equals 1 at the source and decreases in t; the bridge depends on t only through a, and the results below hold for every $a \in (0, 1]$.

B.1 BRIDGE DENSITY

$\mathbb{S}$-FLM draws $\boldsymbol{z}_0$ uniformly on $\mathbb{S}^{d-1}$ and sets $\boldsymbol{z}_t = \mathrm{SLERP}(\boldsymbol{z}_0, \boldsymbol{e}_x; \alpha_t)$. About $\boldsymbol{e}_x$, write $\boldsymbol{z} = \cos\theta\, \boldsymbol{e}_x + \sin\theta\, \boldsymbol{\xi}$ with $\boldsymbol{\xi}$ in the unit sphere $\mathbb{S}^{d-2}$ of the orthogonal complement of $\boldsymbol{e}_x$. The surface measure factorizes as

$$d\sigma(\boldsymbol{z}) = \sin^\nu \theta\; d\theta\; d\sigma_{d-2}(\boldsymbol{\xi}). \tag{20}$$

Let $\omega = \theta_x(\boldsymbol{z}_0)$. Then $\mathrm{SLERP}(\boldsymbol{z}_0, \boldsymbol{e}_x; \alpha) = \big[\sin(a\omega)\, \boldsymbol{z}_0 + \sin((1-a)\omega)\, \boldsymbol{e}_x\big] / \sin\omega$, and substituting $\boldsymbol{z}_0 = \cos\omega\, \boldsymbol{e}_x + \sin\omega\, \boldsymbol{\xi}$ gives $\boldsymbol{z}_t = \cos(a\omega)\, \boldsymbol{e}_x + \sin(a\omega)\, \boldsymbol{\xi}$: SLERP keeps $\boldsymbol{\xi}$ and contracts the polar angle by the factor a. For uniform $\boldsymbol{z}_0$, $(\omega, \boldsymbol{\xi})$ has density $\sin^\nu \omega / |\mathbb{S}^{d-1}|$ with respect to $d\omega\, d\sigma_{d-2}(\boldsymbol{\xi})$. The map $\omega \mapsto a\omega$ has Jacobian a, and dividing by the factor $\sin^\nu\theta$ of Eq. (20) gives

$$q_t(\boldsymbol{z} \mid x) = \frac{\sin^\nu(\theta/a)}{a\, |\mathbb{S}^{d-1}|\, \sin^\nu \theta}\, \mathbf{1}\{\theta < a\pi\}, \tag{21}$$

which is Eq. (15).

B.2 TEMPERED BRIDGE

By Eq. (20), under $q_{t,T}(\cdot \mid x) \propto q_t(\cdot \mid x)^\beta$ the direction $\boldsymbol{\xi}$ is uniform on $\mathbb{S}^{d-2}$ and independent of the polar angle, whose marginal is

$$m_{a,T}(\theta) = \frac{g_{a,T}(\theta)}{C_{a,T}}, \qquad g_{a,T}(\theta) = \sin^{\nu\beta}\!\Big(\frac{\theta}{a}\Big)\, \sin^{\nu(1-\beta)}\theta, \qquad \theta \in (0, a\pi), \tag{22}$$

with $C_{a,T} = \int_0^{a\pi} g_{a,T}$. Rotations preserve σ and map the bridges of different tokens onto one another, so $Z_{t,T}$ depends only on (a, T, d). Since $g_{a,T}$ behaves like θ^ν as $\theta \to 0$ and like $(a\pi - \theta)^{\nu\beta}$ as $\theta \to a\pi$ ($a < 1$), $Z_{t,T} < \infty$ and $q_{t,T}$ is a valid density for every $T > 0$. Its support does not depend on T. At $a = 1$, $g_{1,T} = \sin^\nu\theta$ for every T, so the tempered source $q_{0,T}$ is the uniform source of $\mathbb{S}$-FLM.

B.3 TEMPERED CONDITIONAL VELOCITY

We write the polar dynamics as $d\theta/da = v_{a,T}(\theta)$.

Symmetry reduction. $q_{t,T}(\cdot \mid x)$ is invariant under the orthogonal transformations fixing $\boldsymbol{e}_x$, and we take its conditional velocity to be equivariant, $u(R\boldsymbol{z}) = Ru(\boldsymbol{z})$ for all such R. The transformations fixing both $\boldsymbol{e}_x$ and $\boldsymbol{z}$ therefore fix $u(\boldsymbol{z})$, and for $d \geq 3$ the only tangent vectors at $\boldsymbol{z}$ they fix are multiples of $\mathbf{e}_\theta$, the unit tangent pointing away from $\boldsymbol{e}_x$ along the geodesic. Equivariance under transformations that move $\boldsymbol{\xi}$ makes the magnitude a function of θ alone, so $u = \dot\theta\, \mathbf{e}_\theta$.

One-dimensional continuity equation. For a density ρ with respect to σ, $\nabla \cdot (\rho\, v\, \mathbf{e}_\theta) = \sin^{-\nu}\theta\; \partial_\theta(\sin^\nu\theta\, \rho\, v)$. Multiplying $\partial_a \rho + \nabla \cdot (\rho\, v\, \mathbf{e}_\theta) = 0$ by $|\mathbb{S}^{d-2}| \sin^\nu\theta$, which turns ρ into the polar marginal $m_{a,T}$, gives

$$\partial_a m_{a,T}(\theta) + \partial_\theta\big(m_{a,T}(\theta)\, v_{a,T}(\theta)\big) = 0. \tag{23}$$

With no flux through the pole ($m\, v \to 0$ as $\theta \to 0$), integrating over $(0, \theta)$ gives the unique solution $v_{a,T} = -\partial_a M_{a,T} / m_{a,T}$, where $M_{a,T}(\theta) = \int_0^\theta m_{a,T}$.

Closed form. From Eq. (22), $\partial_a g_{a,T}(\theta) = -(\nu\beta/a^2)\, \theta \cot(\theta/a)\, g_{a,T}(\theta)$. Since $g_{a,T}(a\pi) = 0$, differentiating $C_{a,T}$ produces no boundary term, and $\partial_a C_{a,T} = -(\nu\beta/a^2)\, \bar H_{a,T}\, C_{a,T}$ with $\bar H_{a,T} = \mathbb{E}_{\theta \sim m_{a,T}}[\theta \cot(\theta/a)]$. Hence

$$v_{a,T}(\theta) = \frac{\nu\beta}{a^2}\, \frac{\int_0^\theta \big[\vartheta \cot(\vartheta/a) - \bar H_{a,T}\big]\, g_{a,T}(\vartheta)\, d\vartheta}{g_{a,T}(\theta)}. \tag{24}$$

Velocity field. As t increases, $da/dt = -\dot\alpha_t$, so $\dot{\boldsymbol{z}} = -\dot\alpha_t\, v_{a,T}(\theta)\, \mathbf{e}_\theta$. Since $\mathrm{Log}_{\boldsymbol{z}}(\boldsymbol{e}_x) = -\theta\, \mathbf{e}_\theta$, this equals

$$u_{t,T}(\boldsymbol{z} \mid x) = w_{t,T}(\theta)\, u_t(\boldsymbol{z} \mid x), \qquad w_{t,T}(\theta) = \frac{a_t\, v_{a_t,T}(\theta)}{\theta}, \qquad u_t(\boldsymbol{z} \mid x) = \frac{\dot\alpha_t}{a_t}\, \mathrm{Log}_{\boldsymbol{z}}(\boldsymbol{e}_x),$$

which is Eq. (16). At $T = 1$, integration by parts with $s = \theta/a$ gives $\int_0^\pi s\cos s\,\sin^{\nu-1} s\,ds = -\frac{1}{\nu}\int_0^\pi \sin^\nu s\,ds$, hence $\bar{H}_{a,1} = -a/\nu$, and $\frac{d}{d\theta}\big[\frac{\theta}{a} g_{a,1}(\theta)\big] = \frac{\nu}{a^2}\big[\theta\cot(\theta/a) - \bar{H}_{a,1}\big] g_{a,1}(\theta)$. The integral in Eq. (24) thus equals $\frac{a}{\nu}\,\theta\,g_{a,1}(\theta)$, so $v_{a,1} = \theta/a$ and $w_{t,1} \equiv 1$: u_t is the $\mathbb{S}$-FLM conditional velocity, and the tempered construction reduces to it at $T = 1$.

B.4 Exact sampling of the tempered bridge

Use the normalized angles $\psi = \omega/\pi$ and $\phi = \theta/(a\pi)$ in $(0, 1)$. Let F_{src} be the CDF of the density $\propto \sin^\nu(\pi\psi)$, and let $F_{a,T}$ be the CDF of the density $\propto \sin^{\nu\beta}(\pi\phi)\sin^{\nu(1-\beta)}(a\pi\phi)$, which is Eq. (22) in the variable ϕ. Define

$$Q_{a,T}(\omega) = a\pi\, F_{a,T}^{-1}\big(F_{\mathrm{src}}(\omega/\pi)\big). \tag{25}$$

Under both the source and the tempered bridge, $\boldsymbol{\xi}$ is uniform on $\mathbb{S}^{d-2}$ and independent of the polar angle. Hence $\boldsymbol{z} = \cos\theta\,\boldsymbol{e}_x + \sin\theta\,\boldsymbol{\xi}$ with $\theta = Q_{a,T}(\omega)$ is an exact sample of $q_{t,T}(\cdot \mid x)$. At $\beta = 1$, $F_{a,1} = F_{\mathrm{src}}$ and $\theta = a\omega$, which reproduces SLERP for the same $\boldsymbol{z}_0$. Equivalently, $\boldsymbol{z} = \mathrm{SLERP}(\boldsymbol{z}_0, \boldsymbol{e}_x; 1 - \theta/\omega)$, a SLERP with a position-specific parameter. For TFT we build the target CDF for each (a_t, T) pair in the batch on a grid refined geometrically toward both ends of $[0, a_t\pi]$ and invert it at the drawn angles, with no interpolation table over (a, T).

C Held-out selection of λ

Held-out set. The TinyGSM data used by $\mathbb{S}$-FLM are split into a training part, which the TFT continuation reads, and a validation part of 118,216 problems, which it does not. From a random subset of 768 validation problems, we remove 111 prompts that also occur verbatim in the training part and 30 problems without a reliably recoverable reference answer (seven fall in both groups), leaving 634 problems, none of which matches a GSM8K test problem. The reference answer of each problem is the value returned by its TinyGSM reference program, and generated programs are graded as on GSM8K.

Protocol. For each backbone, the sweep uses the TFT checkpoint, generation seed 1 and the GSM8K sampling protocol. Each $\lambda \in \{0, 0.25, 0.5, 0.75, 1\}$ is scored by its held-out accuracy averaged over $T \in \{0.4, 0.2, 0.1\}$; the maximizer, with ties broken toward the larger λ, is used at every temperature. This selects $\lambda = 0.5$ for S-arch and $\lambda = 0.25$ for DiT (Figure 5, Table 2).

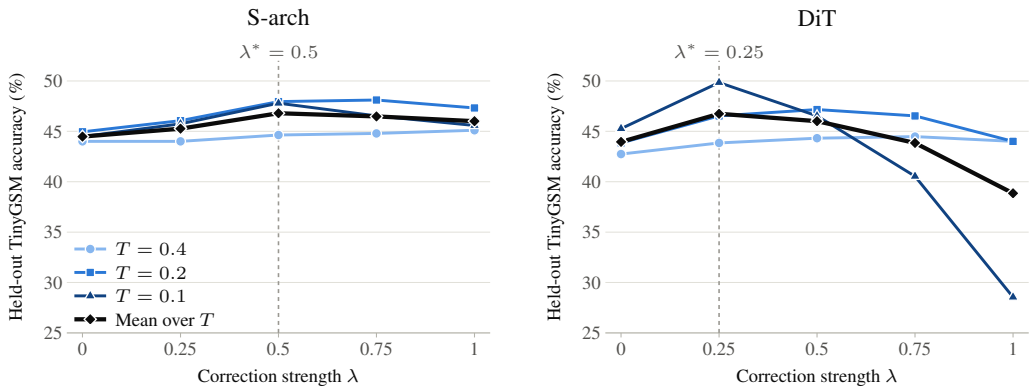

Figure 5: Held-out selection of λ for BCT-L (634 held-out TinyGSM problems, TFT, seed 1). Thin lines: accuracy at each selection temperature; thick line: their mean, the selection score. The dotted line marks the selected λ. $\lambda = 0$ applies no Bayes correction and $\lambda = 1$ the full correction. Values in Table 2.

Table 2: Held-out λ sweep: correct answers out of 634 at each selection temperature and the selection score (mean accuracy over the three temperatures, %); bold: selected value. The last column is a DiT TFT checkpoint after 500 updates, not used for selection.

	S-arch				DiT				DiT (500 upd.)
λ	T=0.4	0.2	0.1	Score	T=0.4	0.2	0.1	Score	Score
0	279	285	282	44.48	271	278	287	43.95	42.27
0.25	279	292	290	45.27	278	295	316	**46.74**	**48.74**
0.5	283	304	303	**46.79**	281	299	295	46.00	48.32
0.75	284	305	295	46.48	282	295	257	43.85	45.79
1	286	300	289	46.00	279	279	181	38.85	43.22

D PASS@K ON S-ARCH

This appendix gives the setup and values behind Section 4.3.

Setup. We compare Cont. with standard tempering and TFT with BCT-L ($\lambda = 0.5$) on S-arch, using the checkpoints, sampler, prompts and grader of Table 1 at the same eight temperatures. The sampler draws one generation per problem, so the eight generations per problem come from eight runs with generation seeds $1 + 10j$, $j = 0, \ldots, 7$; the first three are the seeds of Table 1 and reproduce its generations exactly. For a problem with c correct generations out of $n = 8$, we use the unbiased estimator $1 - \binom{n-c}{k} / \binom{n}{k}$ (Chen et al., 2021), averaged over the 1319 problems. Intervals are 95% percentile bootstrap intervals over problems (10,000 resamples), paired for differences.

Table 3: Pass@K (%) of Figure 2: unbiased estimator over $n = 8$ generations per problem (generation seeds $1, 11, \ldots, 71$), S-arch, 1319 GSM8K problems. Pass@1 is the mean single-generation accuracy over the eight draws and agrees with the three-seed mean of Table 1 to within 0.6 points.

	Cont. standard tempering				TFT BCT-L			
T	$K = 1$	2	4	8	$K = 1$	2	4	8
1	12.3	19.6	28.3	37.5	12.7	20.2	29.0	37.7
0.8	14.4	22.5	32.1	42.0	16.5	25.1	34.7	44.5
0.6	15.8	24.3	34.1	44.5	20.2	29.8	40.3	51.3
0.5	16.9	25.8	36.0	46.8	21.8	31.3	41.2	51.1
0.4	17.4	26.3	36.7	47.5	23.3	32.9	42.8	52.5
0.3	17.6	26.7	36.9	47.3	24.4	33.7	43.2	52.5
0.2	17.7	26.6	36.6	47.2	25.3	34.5	43.4	51.9
0.1	17.8	26.8	37.0	47.5	27.1	35.7	43.9	51.1

Differences. For every $T < 1$ and $K \leq 8$, the paired interval of the difference (BCT-L minus standard tempering) lies above zero. At $T = 0.1$ the difference is 9.2 [8.1, 10.3] points at $K = 1$ and 3.6 [1.4, 5.8] at $K = 8$. The number of problems solved by at least one of the eight generations is 693 and 674 for BCT-L at $T = 0.3$ and 0.1, against 624 and 627 for standard tempering.

E STANDARD TEMPERATURE SCALING ON DISCRETE DIFFUSION BASELINES

We sweep the sampling temperature of the MDLM (Sahoo et al., 2024) and Duo (Sahoo et al., 2025) TinyGSM checkpoints released by Deschenaux & Gulcehre (2026), without any training or tuning. Both use the ancestral samplers of that release: at every step the denoiser logits are divided by T before the softmax, and the tempered clean-token distribution enters the unchanged reverse transition (absorbing state for MDLM, uniform state for Duo). At $T = 1$ this reproduces the released sampler exactly (1319 of 1319 identical programs for both models, checked with 32 steps). We use 1024 steps, generation seed 1 and the GSM8K evaluation of Section 4.1; the $\mathbb{S}$-FLM values are the standard-tempering results at seed 1.

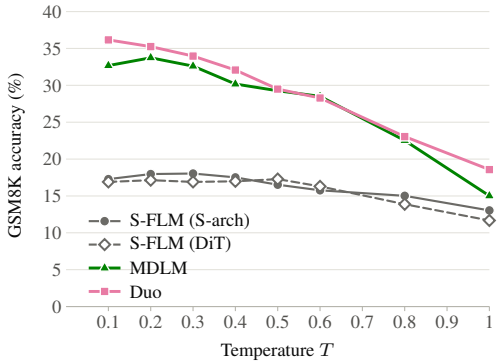

Figure 6: Standard temperature scaling on the TinyGSM baselines released with $\mathbb{S}$-FLM (NFE $=$ 1024, seed 1). Gray: $\mathbb{S}$-FLM (Section 4.2). Values in Table 4.

Table 4: Accuracy (%) and correct answers out of 1319 for Figure 6. $\mathbb{S}$-FLM uses the exact velocity.

T	MDLM	Duo	$\mathbb{S}$-FLM (S-arch)	$\mathbb{S}$-FLM (DiT)
1	15.0 (198)	18.6 (245)	13.0 (172)	11.7 (154)
0.8	22.5 (297)	23.0 (304)	15.0 (198)	13.9 (183)
0.6	28.5 (376)	28.3 (373)	15.8 (208)	16.3 (215)
0.5	29.3 (386)	29.5 (389)	16.5 (218)	17.3 (228)
0.4	30.2 (398)	32.1 (423)	17.5 (231)	17.0 (224)
0.3	32.6 (430)	34.0 (448)	18.0 (238)	16.9 (223)
0.2	33.7 (445)	35.3 (465)	18.0 (237)	17.1 (226)
0.1	32.7 (431)	36.2 (477)	17.3 (228)	16.9 (223)

At $T = 0.1$, MDLM and Duo solve 32.7% and 36.2%, in line with the 33% and 36% reported by Deschenaux & Gulcehre (2026); at $T = 1$ they solve 15.0% and 18.6%, against the reported 18.0% and 17.2%.

F SUDOKU-HARD

This appendix gives the setup and values behind Section 4.4.

Task. We use the Sudoku-Hard task and puzzle generator of $\mathbb{S}$-FLM (Deschenaux & Gulcehre, 2026) unchanged. Every puzzle has 30 clues and a unique solution (checked with a solution counter), and the 48,000 training and 2,000 test puzzles share no puzzle or solution. A sequence is the puzzle followed by its solution, each written as 81 digits with row separators (a vocabulary of 12 tokens). The puzzle is a fixed prefix during sampling, and a test puzzle counts as solved only if the generated solution equals the reference.

Model and training. We train with the released Sudoku recipe of $\mathbb{S}$-FLM unchanged: a DiT denoiser with 8 blocks and hidden size 512, the geodesic bridge with its truncated adaptive noise schedule, 20,000 updates with a batch of 256 puzzles on one GPU, AdamW with a constant learning rate of 3×10^{-4} after warmup, EMA decay 0.9999, and training seed 1. Our code reproduces the released code bit for bit on this recipe (weights, EMA weights and optimizer state after 1,200 updates). We call the result the base checkpoint. Cont. and TFT resume its complete training state for 2,000 updates; TFT uses the TFT setting of Section 4.1 with T-conditioning. Each checkpoint is a single training run.

Evaluation. All samplers use the released Sudoku sampler (180 NFE, exact velocity, argmax decoding, EMA weights, batches of 64 puzzles on one GPU) with generation seeds 1, 11 and 21; only the temperature and the BCT rule change between runs, and for a given seed all samplers start from the same noise. At $T = 1$ the training-free samplers coincide up to floating-point differences, which change the number of solved puzzles by at most 2 per seed.

Selection of λ. We select λ with the protocol of Appendix C on 2,000 new puzzles from the same generator with a different seed, disjoint from the training and test puzzles. Held-out accuracy increases with λ at every selection temperature (Table 6), so $\lambda = 1$.

Table 5: Sudoku-Hard exact solve rate (%), mean $\pm$ standard deviation over generation seeds 1, 11 and 21 (2,000 test puzzles, NFE $= 180$). BCT-L uses the held-out $\lambda = 1$. The seed standard deviation is dominated by a shift shared by all samplers.

	Base checkpoint			Cont.			TFT
T	Standard	BCT-E	BCT-P	Standard	BCT-E	BCT-P	BCT-L
1	42.9 ± 1.5	42.9 ± 1.6	42.9 ± 1.5	41.7 ± 1.2	41.7 ± 1.2	41.7 ± 1.2	41.7 ± 1.1
0.8	42.6 ± 1.6	43.7 ± 1.4	43.6 ± 1.0	41.6 ± 1.2	42.6 ± 1.2	42.4 ± 1.0	42.7 ± 0.7
0.6	42.3 ± 1.3	44.1 ± 1.2	44.2 ± 1.1	41.4 ± 1.1	43.1 ± 1.0	42.7 ± 1.2	44.3 ± 1.2
0.5	42.2 ± 1.3	44.4 ± 1.3	44.4 ± 1.2	41.3 ± 1.0	43.3 ± 1.2	43.0 ± 0.8	45.1 ± 1.3
0.4	42.1 ± 1.2	44.8 ± 1.4	44.8 ± 1.1	41.2 ± 1.0	43.5 ± 1.2	43.2 ± 0.5	45.6 ± 1.3
0.3	42.0 ± 1.2	45.0 ± 1.4	44.7 ± 0.9	41.2 ± 1.0	43.7 ± 1.2	43.3 ± 0.5	45.9 ± 1.1
0.2	41.9 ± 1.3	45.2 ± 1.4	44.8 ± 0.4	41.2 ± 1.0	44.0 ± 1.2	43.6 ± 0.1	46.4 ± 1.1
0.1	41.7 ± 1.3	45.5 ± 1.4	44.9 ± 0.4	41.0 ± 1.0	44.2 ± 1.4	43.7 ± 0.5	46.8 ± 1.2

Table 6: Held-out selection of λ for BCT-L on Sudoku-Hard: exact solve rate (%) of TFT on 2,000 held-out puzzles (generation seed 1, NFE $= 180$). Mean: the selection score. Bold: selected. No test puzzle was used.

λ	$T = 0.4$	$T = 0.2$	$T = 0.1$	Mean
0	41.8	40.8	40.1	40.9
0.25	42.8	43.5	43.3	43.2
0.5	44.0	44.5	44.9	44.5
0.75	44.9	45.4	46.9	45.7
1	**45.6**	**46.3**	**47.5**	**46.5**

G BCT FOR vMF FLOWS

This appendix gives the instantiation and setup of Section 4.5. SFCD (Chemseddine et al., 2026) keeps learned token embeddings on the unit sphere but uses the von Mises–Fisher bridge $q_t(\boldsymbol{z} \mid x) = \mathrm{vMF}(\boldsymbol{e}_x, \kappa_t)$ with increasing concentration and $\kappa_0 = 0$. Its sampler integrates the posterior-induced velocity Eq. (2) on a time warp chosen by the release. We use the released TinyGSM checkpoint (250k updates) with its architecture, warp and sampler unchanged, and change only the velocity.

Tempering the vMF bridge. The tempered bridge is $\mathrm{vMF}(\boldsymbol{e}_x, \kappa_t/T)$ Eq. (18), whose normalizer does not depend on x, so BCT-P keeps the posterior weights of standard tempering. Because the vMF bridge is indexed by its concentration, the released conditional velocity has the form $u_t(\boldsymbol{z} \mid x) = \dot{\kappa}_t \, V_{\kappa_t}(\boldsymbol{z} \mid x)$, where $V_\kappa(\cdot \mid x)$ transports $\mathrm{vMF}(\boldsymbol{e}_x, \kappa)$ as κ increases. The tempered bridge is the same family at concentration κ_t/T, so its conditional velocity is

$$u_{t,T}(\boldsymbol{z} \mid x) = \frac{\dot{\kappa}_t}{T} \, V_{\kappa_t/T}(\boldsymbol{z} \mid x) : \tag{26}$$

the released velocity evaluated at concentration κ_t/T, with the concentration advancing $1/T$ times as fast. BCT-P and BCT-L integrate this velocity on the warp grid of the release. BCT-E keeps the original bridge and dynamics and, since $\log q_t(\boldsymbol{z} \mid x) = \kappa_t \langle \boldsymbol{e}_x, \boldsymbol{z} \rangle$ up to a constant in x, adds $(1-\beta)\,\kappa_t \langle \boldsymbol{e}_x, \boldsymbol{z} \rangle$ to the tempered logits. At $T = 1$ every variant reduces to the released sampler.

TFT and BCT-L. TFT uses the parameterization of Section 3.3: the network outputs a remainder a_θ, the posterior is $\mathrm{softmax}(\boldsymbol{a}_\theta + \boldsymbol{g})$ with the analytic term $g_x = (\kappa_t/T)\langle \boldsymbol{e}_x, \boldsymbol{z} \rangle$, and training uses the cross-entropy of Eq. (6). The temperature enters the network through $\kappa_t/(T\kappa_{\max})$. BCT-L Eq. (9) then takes the form $\ell_x = \beta \, a_{\theta,x} + [\beta + \lambda(1-\beta)] \, g_x$.

Continued training. Both continuations resume the released checkpoint for 20,000 updates: Cont. with the original recipe (with the warp frozen, as in TFT) and TFT with the temperature distribution of Section 4.1 and latents drawn from $q_{t,T}$. Both are evaluated with an EMA accumulated over the continuation (decay 0.9999); the released checkpoint uses its released EMA weights. Each continuation is a single training run.

Evaluation. We use the released SFCD evaluation of the 1319 GSM8K test problems (warp-aware ODE sampler, NFE $= 1024$, batches of 32, and its own answer checker) with generation seeds 42–46; every cell of Table 7 is complete over the five seeds. The separate $T = 1$ runs of BCT-E on the released checkpoint reproduce standard tempering exactly.

Selection of λ. We select λ with the protocol of Appendix C on 1024 TinyGSM validation problems that the continuation did not read and that share no problem with the GSM8K test set. The score increases with λ (51.92, 53.19, 53.87, 54.07 and 54.39% for $\lambda = 0, 0.25, 0.5, 0.75, 1$), so $\lambda = 1$.

Table 7: BCT on SFCD: GSM8K accuracy (%), mean $\pm$ sample standard deviation over generation seeds 42–46 (1319 problems, NFE $= 1024$). Released: the SFCD TinyGSM checkpoint with its released EMA weights. Cont. and TFT: 20,000 updates from it, evaluated with an adaptation-local EMA. BCT-L uses the held-out $\lambda = 1$. At $T = 1$ every correction is the identity, so the three columns of a model coincide.

	Released			Cont.			TFT
T	Standard	BCT-E	BCT-P	Standard	BCT-E	BCT-P	BCT-L
1	23.55 ± 0.79	23.55 ± 0.79	23.55 ± 0.79	23.59 ± 0.71	23.59 ± 0.71	23.59 ± 0.71	21.55 ± 0.86
0.8	27.79 ± 0.83	29.19 ± 1.43	29.39 ± 0.48	27.99 ± 0.27	29.80 ± 0.75	29.28 ± 0.74	28.19 ± 0.97
0.6	30.34 ± 0.63	31.71 ± 0.64	31.63 ± 0.77	30.61 ± 0.77	31.54 ± 0.78	32.43 ± 0.45	32.05 ± 0.47
0.5	30.80 ± 0.80	31.40 ± 0.62	32.36 ± 1.03	31.80 ± 0.77	31.21 ± 0.47	33.28 ± 0.39	33.54 ± 0.63
0.4	31.19 ± 0.99	28.26 ± 0.52	33.39 ± 0.90	32.57 ± 0.83	26.37 ± 0.32	33.93 ± 0.58	34.33 ± 0.97
0.3	31.75 ± 0.91	15.30 ± 0.67	33.63 ± 0.69	32.62 ± 0.61	2.93 ± 0.25	33.54 ± 0.75	34.47 ± 0.55
0.2	31.71 ± 0.40	0.08 ± 0.11	32.05 ± 0.67	32.16 ± 0.40	0.02 ± 0.03	32.24 ± 0.28	33.24 ± 1.12
0.1	30.99 ± 0.55	0.00 ± 0.00	26.63 ± 0.85	31.25 ± 0.42	0.00 ± 0.00	28.49 ± 0.54	31.68 ± 0.53