SINGLE-MODEL ENSEMBLING VIA TEST-TIME STOCHASTIC ENSEMBLES

Anonymous authors
Paper under double-blind review

ABSTRACT

Ensemble methods are a well-established paradigm of machine learning and have recently emerged as a successful technique for obtaining inference-time gains in deep neural networks. In contrast to prior work, we create a **test-time stochastic ensemble (TTSE)** from a *single* transformer by generating multiple stochastic forward passes from a frozen checkpoint and aggregating their outputs, without any additional training, checkpoints, or weight updates. We carefully investigate the precise design of the stochasticity and ensembling mechanism, finding that (1) adding isotropic Gaussian noise to Query and Key vectors is the most effective; (2) averaging probabilities is better than averaging logits; and (3) increasing the sample size enhances the differentiability of correct reasoning traces from incorrect ones. Across next-token prediction and reasoning tasks, our method consistently improves upon the deterministic-inference baseline and alternative stochastic methods, suggesting that *successful ensemble methods do not actually require training multiple models, which are very expensive in practice*.

1 INTRODUCTION

Ensemble methods, which systematically combine multiple models or outputs to improve performance and robustness, have a history dating back to Dasarathy & Sheela (1979) and Schapire (1990). When applied to modern neural networks, successful inference-time gains have typically come from *training-time ensembling*, in which the ensemble structure is accounted for during training and incorporated into the final parameters θ. A number of training-time ensemble methods have been proposed, including deep ensembles (Lakshminarayanan et al., 2017), stochastic-depth networks (Huang et al., 2016), SWAG (Maddox et al., 2019), weight averaging (Izmailov et al., 2018), and stochastic attention (Pei et al., 2022). The complementary regime, which we call *test-time ensembling*, combines stochastic variations of a single model or multiple independently trained models during the inference phase without requiring the ensemble structure to be explicitly defined or optimized during training. This line of work has been explored by methods such as MC Dropout (Gal & Ghahramani, 2016), test-time augmentation (Shanmugam et al., 2021), and self-consistency (Wang et al., 2023). While powerful, training-time ensembling is limited in practice by the structural cost of ensemble-aware training, requiring prohibitive amounts of compute to retrain or train from scratch. Similarly, test-time ensembling of multiple models necessitates several independently trained models, which is an increasingly infeasible demand as model sizes continue to scale.

In light of these limitations, we propose **test-time stochastic ensembling (TTSE)**, a training-free and tuning-free method that uses a *single* transformer to generate a test-time ensemble, thereby circumventing the impractical cost of other ensemble methods. Since the output logits of a standard transformer are deterministic when given the same input, TTSE introduces stochasticity at inference time to obtain distinct outputs. After drawing N stochastic forward passes, we then aggregate the per-pass logits to form the final output. Notably, the overhead cost of TTSE is paid only at inference time. Moreover, the base model can be any sufficiently pretrained model, and the mechanism is fully specified by four design choices: the noise location, the noise distribution, the aggregator, and the sample size.

To navigate the infinite space of possible configurations, we systematically vary each of the four axes to identify an effective TTSE method, which we call **stochastic attention ensembling (SAE)**. SAE consists of only a few lines of code, requires no retraining, and does not include additional parameters

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

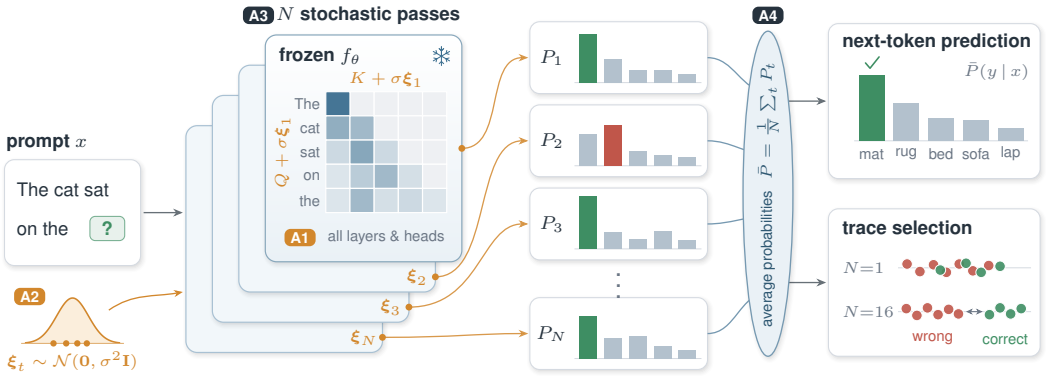

Figure 1: **Stochastic attention ensembling (SAE)**, shown on a toy example. A single frozen transformer f_θ is run N times on the same input. In every pass, i.i.d. isotropic Gaussian noise $\sigma\boldsymbol{\xi}_t$ (orange) is added to the Query and Key vectors of every layer and head, which perturbs the attention pattern. A single pass can be wrong (red bar in P_2), but averaging the N predictive distributions recovers the correct next token (green). The same per-pass probabilities also score candidate reasoning traces, and correct traces separate from wrong ones as N grows. A1–A4 mark the four design axes of TTSE (Section 2.2). No extra checkpoints, retraining, or weight updates are needed.

(Figure 1). For next-token prediction using a 2.7B-parameter GPT trained on FineWeb (Penedo et al., 2024), SAE reduces cross-entropy validation loss by 0.013 compared to the deterministic baseline; for comparison, this improvement is $13\times$ and $15\times$ larger than those of MC Dropout and weight perturbation, respectively. Furthermore, SAE's advantage scales *positively* with model size, being $6.4\times$ larger on the 2.7B model than on a 300M model with the same experimental setup.

TTSE methods also demonstrate superior reasoning capabilities. On 360 problems across 3 models (DeepSeek-R1-0528-Qwen3-8B (DeepSeek-AI et al., 2025; Qwen Team, 2025), Qwen3-32B (Qwen Team, 2025), and Qwen3-30B-A3B-Thinking-2507 (Qwen Team, 2025)) and 4 math competitions (BRUMO25 (Brown University Mathematical Olympiad, 2025), AIME24 (Mathematical Association of America, 2024), AIME25 (Mathematical Association of America, 2025), and HMMT25 (Harvard-MIT Mathematics Tournament, 2025)), SAE-based trace selection correctly answers 7 more problems than DeepConf's (Fu et al., 2026) strongest baseline. We identify a specific phenomenon that enables these gains: SAE separates correct reasoning traces from incorrect ones in log-probability space, with this effect becoming increasingly apparent as the sample size grows (Figure 2).

Our main contributions are as follows:

- We introduce test-time stochastic ensembling (TTSE) as a distinct framework within the test-time ensembling regime, characterized by four axes that encapsulate a wide range of prior methods and establish a foundation for further research.
- We provide an explicit instance of the TTSE framework, stochastic attention ensembling (SAE), and conduct extensive ablation experiments to show that it consistently outperforms alternative design choices across different model scales, noise locations, noise scales, sample sizes, and aggregation mechanisms.
- We demonstrate the effectiveness of SAE on pretraining for next-token prediction by showing that it reliably decreases validation loss compared to the deterministic baseline and other methods within the TTSE framework.
- We show that SAE improves reasoning-trace selection across different scoring rules. For log-probability scores in particular, we attribute this improvement to SAE progressively enlarging the gap between correct traces and incorrect ones as the sample size increases.

We discuss related work in Appendix A.

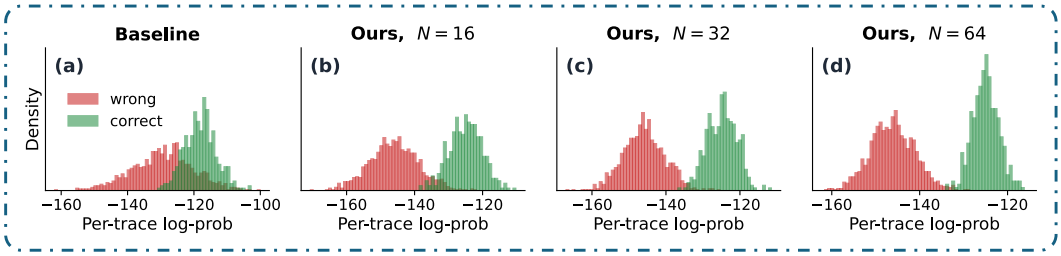

Figure 2: Per-trace log-probability distributions for correct (green) and wrong (red) reasoning traces for BRUMO25 P30 / DeepSeek-R1-Qwen3-8B on the baseline and SAE (Ours) with three different sample sizes. *(a) Baseline:* the two distributions overlap considerably. *(b)–(d) SAE (ours) with* $N \in \{16, 32, 64\}$ *and* $\sigma = 0.5$*:* the two distributions become progressively disjoint as the sample size N increases.

2 Test-Time Stochastic Ensembles

2.1 Definition

Test-time stochastic ensemble (TTSE). Given a pretrained model f_θ with frozen parameters θ, a *test-time stochastic ensemble* is a pair $(\mathcal{P}, \mathcal{A})$, where $\mathcal{P}$ is a stochastic sampler that, given input x and noise $\boldsymbol{\xi}$, produces a single forward-pass output $f_\theta(x; \boldsymbol{\xi})$, and $\mathcal{A}$ is an aggregation mechanism that combines the outputs $\{f_\theta(x; \boldsymbol{\xi}_t)\}_{t=1}^N$ into a final output $\bar{y}$. The parameters θ are not updated or retrained, and the only design choices are those of $\mathcal{P}$ and $\mathcal{A}$. This framework subsumes a wide range of prior work and is broad enough to provide new directions for exploration.

Training-time ensembles. It is worth noting the distinction between test-time ensembles and training-time ensembles. A *training-time ensemble* embeds the ensemble structure within θ via training, e.g. deep ensembles (Lakshminarayanan et al., 2017) train N separate networks and average their predictions; stochastic depth (Huang et al., 2016) and hierarchical stochastic attention (Pei et al., 2022) apply noise during training and modify the optimization landscape; SWAG (Maddox et al., 2019) learns a posterior over θ as part of training. In this regime, the cost of ensembling is paid during training, and the resulting model is no longer a single pretrained checkpoint. On the other hand, a TTSE holds θ fixed and obtains diversity entirely from a noise process applied at inference.

2.2 The Four Design Axes

Despite the strict constraint of keeping the pretrained model parameters fixed, our framework is naturally characterized by a large design space that has not been fully explored by prior literature. In particular, a TTSE is completely specified by four axes that define $\mathcal{P}$ and $\mathcal{A}$:

- (A1) the *location* at which noise is added during inference, e.g. the input x, the embedding layer, the weights θ, intermediate activations, the attention scores or values, the output logits. The choice is typically structured, e.g. per-layer, per-head, per-token.
- (A2) the *distribution* of the noise, which is characterized by distribution parameters (e.g. variance) and can be discrete (e.g. Bernoulli, dropout) or continuous (e.g. Gaussian, uniform).
- (A3) the *sample size*, i.e. the number of stochastic forward passes N.
- (A4) the *aggregation mechanism*, e.g. those listed in Table 3 for next-token prediction and Table 4 for reasoning-trace selection.

Relationship with existing methods. Although not strictly examples of TTSE, prior methods have explored these design axes in various capacities. Self-consistency (Wang et al., 2023) fixes (A4) to majority vote of decoded answers, ignoring the per-pass distributions. MC Dropout (Gal & Ghahramani, 2016) fixes (A1) to dropout activations and (A2) to Bernoulli, providing almost no diversity on modern low-dropout LMs. Test-time augmentation (Shanmugam et al., 2021) fixes (A1) to the input. Stochastic depth (Huang et al., 2016) fixes (A1) to layer drops. SWAG (Maddox et al.,

Algorithm 1 Stochastic attention ensembling (SAE) for next-token prediction

Require: Pretrained model f_θ, input x, noise scale σ, sample size N
$\bar{P} \leftarrow \mathbf{0}$ ▷ accumulated probability vector
for $t = 1, \dots, N$ **do**
  Sample $\boldsymbol{\xi}^q_{l,h,i}, \boldsymbol{\xi}^k_{l,h,j} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ for all layers l, heads h, positions i, j
  $\boldsymbol{\xi}_t \leftarrow \{\boldsymbol{\xi}^q_{l,h,i}, \boldsymbol{\xi}^k_{l,h,j}\}_{l,h,i,j}$
  $\bar{P} \leftarrow \bar{P} + \mathrm{softmax}(f_\theta(x; \boldsymbol{\xi}_t))$ ▷ forward pass with noise using (2)
end for
return $\bar{P}/N$ ▷ averaged probabilities

Algorithm 2 Stochastic attention ensembling (SAE) for reasoning-trace selection

Require: Trace pool $\mathcal{T}$, pretrained model f_θ, scoring rule ϕ, aggregator $\mathcal{A}$, noise scale σ, sample size N
for each $t \in \mathcal{T}$ **do**
  for $k = 1, \dots, N$ **do**
    Sample $\boldsymbol{\xi}^q_{l,h,i}, \boldsymbol{\xi}^k_{l,h,j} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ for all layers l, heads h, positions i, j
    $\boldsymbol{\xi}^{(k)} \leftarrow \{\boldsymbol{\xi}^q_{l,h,i}, \boldsymbol{\xi}^k_{l,h,j}\}_{l,h,i,j}$
  end for
  $\bar{\phi}(t) \leftarrow \frac{1}{N} \sum_{k=1}^{N} \phi\big(t \mid f_\theta\big(\cdot; \boldsymbol{\xi}^{(k)}\big)\big)$ ▷ averaged Stage 1 score
end for
return $\mathcal{A}\big(\{(t, \bar{\phi}(t)) : t \in \mathcal{T}\}\big)$ ▷ Stage 2 aggregator

2019) fixes (A1) to weights via a learned posterior. We refer to Section A for an extended discussion of related work.

2.3 Stochastic Attention Ensembling

We now provide an explicit method that, as a quintessential example of the TTSE framework, will be the focus of this work. (A1) is chosen as the Query and Key vectors of every layer, every head, and every position after RoPE and RMSNorm; (A2) is the isotropic zero-centered Gaussian distribution with scale σ typically set to 0.5; (A3) N ranges from 5 to 128; and (A4) is either probability averaging for next-token prediction tasks or an aggregated scoring rule for reasoning-trace selection.

Concretely, given a pretrained transformer with L layers and H attention heads per layer, each head computes the Query, Key, and Value vectors

$$\mathbf{q}_i = \mathrm{RMSNorm}(\mathrm{RoPE}_i(\mathbf{W}_q \mathbf{x}_i)), \quad \mathbf{k}_j = \mathrm{RMSNorm}(\mathrm{RoPE}_j(\mathbf{W}_k \mathbf{x}_j)), \quad \mathbf{v}_j = \mathbf{W}_v \mathbf{x}_j, \tag{1}$$

and forms $\mathbf{y}_i = \sum_j \widetilde{\mathbf{A}}_{ij} \mathbf{v}_j$, where

$$\widetilde{\mathbf{A}}_{ij} = \mathrm{softmax}_j\left(\frac{(\mathbf{q}_i + \sigma\boldsymbol{\xi}^q_i)^\top (\mathbf{k}_j + \sigma\boldsymbol{\xi}^k_j)}{\sqrt{d}}\right), \qquad \boldsymbol{\xi}^q_i, \boldsymbol{\xi}^k_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I}_d). \tag{2}$$

is a perturbation of the standard attention mechanism (Vaswani et al., 2017). We run N forward passes in this fashion and either average the resulting softmax probabilities or aggregate the resulting scores. This procedure, which we call *stochastic attention ensembling (SAE)*, is summarized by Algorithms 1 and 2.

3 Experiments

We evaluate SAE across next-token prediction on FineWeb and reasoning-trace selection on math competitions, including an extensive suite of ablation studies designed to isolate the contribution of each of the design axes in Section 2.2. We further include several scaling and sensitivity analyses that characterize high-performing regimes and identify potential for deployment in state-of-the-art models.

Table 1: Ten candidates for (A1), partitioned into non-attention-level and attention-level noise. $\boldsymbol{\xi}$ and η represent standard Gaussian noise, $\odot$ denotes the Hadamard (elementwise) product, $\mathbf{m}$ is a Bernoulli mask, g is a Gumbel sample, and τ_t is a per-pass temperature.

	Location	Distribution	Perturbation
Non-attention-level			
(a)	input embeddings $\mathbf{x}_i$ (Shanmugam et al., 2021)	Gaussian	$\widetilde{\mathbf{x}}_i = \mathbf{x}_i + \sigma\boldsymbol{\xi}_i$
(b)	intermediate activations (Gal & Ghahramani, 2016)	Bernoulli	$\widetilde{\mathbf{h}}_l = \mathbf{h}_l \odot \mathbf{m}_l$
(c)	weight matrices $\boldsymbol{\theta}$ (Maddox et al., 2019)	Gaussian	$\widetilde{\boldsymbol{\theta}} = \boldsymbol{\theta} + \sigma\boldsymbol{\xi}$
(d)	Value vectors $\mathbf{v}_j$	Gaussian	$\widetilde{\mathbf{v}}_j = \mathbf{v}_j + \sigma\boldsymbol{\xi}_j$
(e)	output logits z (temperature ensembling)	Uniform	$\widetilde{z} = z/\tau_t, \;\; \tau_t \sim \mathcal{U}[0.8, 1.2]$
Attention-level			
(f)	attention pre-softmax logits s_{ij}	Gaussian	$\widetilde{s}_{ij} = s_{ij} + \sigma\eta_{ij}$
(g)	attention pre-softmax logits s_{ij} (Pei et al., 2022)	Gumbel	$\widetilde{s}_{ij} = s_{ij} + \beta g_{ij}$
(h)	Query vectors $\mathbf{q}_i$ only	Gaussian	$\widetilde{\mathbf{q}}_i = \mathbf{q}_i + \sigma\boldsymbol{\xi}^q_i$
(i)	Key vectors $\mathbf{k}_j$ only	Gaussian	$\widetilde{\mathbf{k}}_j = \mathbf{k}_j + \sigma\boldsymbol{\xi}^k_j$
(j)	**Q + K vectors after RoPE/RMSNorm** (SAE)	Gaussian	$\widetilde{\mathbf{q}}_i = \mathbf{q}_i + \sigma\boldsymbol{\xi}^q_i; \;\; \widetilde{\mathbf{k}}_j = \mathbf{k}_j + \sigma\boldsymbol{\xi}^k_j$

Table 2: Comparison of noise location candidates on the 2.7B GPT model. (A2) uses the best distribution parameters for each candidate, (A3) sets $N = 50$, and (A4) is probability averaging. The deterministic baseline loss is 3.2954. Results are sorted from best to worst.

Noise location	Best parameters	Val loss	Δ
Q + K vectors	$\sigma = 0.5$	**3.2821**	$\mathbf{-0.0134}$
Attention logits (Gumbel)	$\beta = 0.5, \mu = 0.0$	3.2831	-0.0123
Attention logits (Gaussian)	$\sigma = 0.7$	3.2835	-0.0119
V vectors	$\sigma = 0.3$	3.2854	-0.0100
K vectors only	$\sigma = 0.5$	3.2859	-0.0095
Q vectors only	$\sigma = 0.5$	3.2877	-0.0078
Temperature ensembling	$\tau \in [0.8, 1.2]$	3.2934	-0.0020
MC Dropout	$p = 0.01$	3.2944	-0.0010
Weight perturbation	$\sigma = 0.05$	3.2945	-0.0009
Embedding noise	$\sigma = 0.3$	3.2951	-0.0003

3.1 Experimental Setup

Next-token prediction. The primary model is a 2.7B-parameter GPT (30 layers, 16 heads, $d_h = 128$, RMSNorm (Zhang & Sennrich, 2019), RoPE (Su et al., 2024), SwiGLU (Shazeer, 2020), U-Net skips) trained on 100M FineWeb (Penedo et al., 2024) tokens (Vegesna et al., 2026); we also test a 300M model (16 layers, 8 heads). Validation loss is reported on 10M tokens. The hardware consists of a 8×H100 setup with Flash Attention 3 (Shah et al., 2024).

Reasoning-trace selection. We evaluated three open-weight reasoning models—DeepSeek-R1-0528-Qwen3-8B (DeepSeek-AI et al., 2025; Qwen Team, 2025) (8B dense), Qwen3-32B (Qwen Team, 2025) (32B dense, TP=2), and Qwen3-30B-A3B-Thinking-2507 (Qwen Team, 2025) (MoE; 30B total / 3.3B active / 128 experts / 8 routed)—on four math competitions: BRUMO25, AIME24, AIME25, HMMT25 (30 problems each, 12 model×dataset combinations, 360 problems total). For each problem, we generated a pool of reasoning traces with DeepConf's online pipeline (Fu et al., 2026) (256-trace budget, confidence-based early stopping) and apply SAE post hoc with $\sigma = 0.5$ and $N = 16$ for each trace. Note that the MoE model uses a separate `Qwen3MoeAttention` class; SAE must patch both `Qwen3Attention` and `Qwen3MoeAttention` (see Appendix H). DeepConf's strongest baseline (best-of-7 confidence variants) is used as the comparison point. All reported numbers in this paper are averages over 64 independent runs per (model, problem) pair (different RNG seeds for both the per-trace generation and the SAE noise samples), unless explicitly stated otherwise.

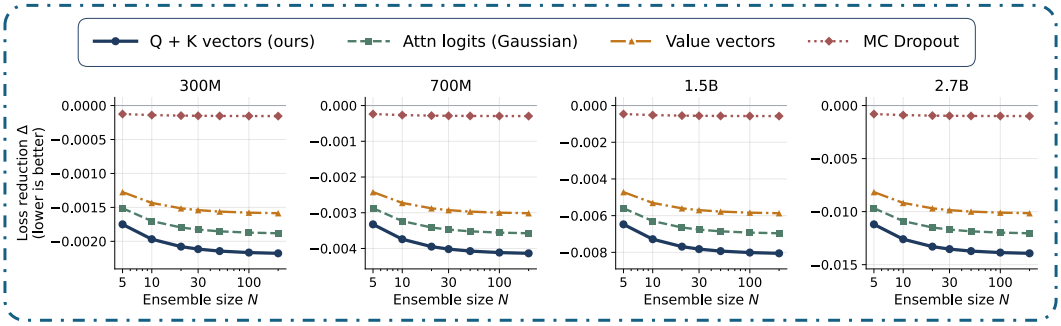

Figure 3: Loss reduction Δ vs. sample size N for four candidate noise locations from Table 1 and seven values of N, on four transformer sizes (300M, 700M, 1.5B, 2.7B). Lower Δ is better. Q + K Gaussian noise (SAE, blue) is the lowest curve at every model size, and the gap between SAE and other candidates widens with scale.

3.2 Next-Token Prediction

We perform a series of extensive ablation studies on the four design axes defined in Section 2.2. The results show that TTSE almost always improves upon the base model for next-token prediction; moreover, SAE achieves better performance than numerous alternative TTSE methods, which justifies its particular design choice.

Ablation on (A1). Among the ten candidate locations for noise injection shown in Table 1, the combination of Query and Key vectors results in the largest improvement over the deterministic baseline (Table 2). In general, we note that attention-level noise tends to yield better performance than non-attention-level noise.

Ablation on (A2) and (A3). Sweeping $\sigma \in [0.1, 0.8]$ and $N \in \{5, 10, 20, 30, 50, 100\}$ shows a clear optimum at $\sigma = 0.5$, regardless of the sample size (Table 6 in Appendix B). Adaptive noise (per-head, per-layer, attention-entropy-based) and temperature scaling provide no further gain: RMS normalization standardizes Query and Key vector magnitudes, leaving little signal for adaptation.

Ablation on (A1) and (A3). Our most important result for next-token prediction concerns the scaling behavior of TTSE. Figure 3 plots the loss reduction Δ as a function of ensemble size N across four model sizes (300M $\rightarrow$ 2.7B) and four representative noise location candidates from Table 1: the Q + K Gaussian noise of SAE, Gaussian noise on attention pre-softmax logits, Value vector Gaussian noise, and MC Dropout (i.e. intermediate activations). Two patterns are visible at every model size. First, the Q + K noise of SAE produces the best Δ at every value of N, amounting to a roughly $14\times$ gain compared to MC Dropout at $N = 50$ and 2.7B. Second, every curve descends further as model size grows—the SAE curve goes from $\Delta \approx -0.002$ at 300M to $\Delta \approx -0.014$ at 2.7B, a $6.4\times$ improvement. Our results suggest that TTSE methods scale with the model size and are therefore well-suited to be carried forward into larger pretrained transformers.

Ablation on (A4). We first compare probability vs. logit averaging, i.e.

$$\bar{P}^{\text{prob}} := \frac{1}{N}\sum_{t=1}^{N} \text{softmax}(\mathbf{z}_t) \qquad \text{v.s.} \qquad \bar{P}^{\text{logit}} := \text{softmax}\left(\frac{1}{N}\sum_{t=1}^{N} \mathbf{z}_t\right),$$

where $\mathbf{z}_t$ denotes the logit vector from sample t. In general, these distributions are not equal due to the nonlinearity of the softmax function. We find that probability averaging outperforms logit averaging by $\approx 2\times$ across several choices of noise (Table 7 in Appendix C). Furthermore, we find that switching a 14-model ensemble's aggregator from logit averaging to probability averaging yields a 0.0095 loss reduction at zero additional compute (Table 8 in Appendix C).

Beyond comparing probabilities vs. logits for aggregation, we sweep the aggregators listed in Table 3, which include several robust aggregators and members of the power mean family M_r. The arithmetic

Table 3: Aggregation rule sweep for SAE with $\sigma = 0.5$ and $N = 50$ on 2.7B GPT. Δ is the change in validation loss vs. the deterministic baseline (3.2954); lower is better. Results are sorted from best to worst.

Aggregator	**Definition**	**Δ**
Arithmetic mean ($r = 1$)	$\bar{P}_y = \frac{1}{N}\sum_t P_y^{(t)}$	$\mathbf{-0.0134}$
Trimmed mean (10%)	drop top/bottom 10% before averaging	-0.0128
Power mean ($r = 2$)	$\bar{P}_y \propto \left(\frac{1}{N}\sum_t (P_y^{(t)})^2\right)^{1/2}$	-0.0119
Per-coordinate median	$\bar{P}_y \propto \mathrm{median}_t\ P_y^{(t)}$	-0.0118
Power mean ($r = 0.5$)	$\bar{P}_y \propto \left(\frac{1}{N}\sum_t (P_y^{(t)})^{1/2}\right)^2$	-0.0105
Geometric mean ($r \to 0$, logit averaging)	$\bar{P}_y \propto \prod_t (P_y^{(t)})^{1/N}$	-0.0070
Harmonic mean ($r = -1$)	$\bar{P}_y \propto \left(\frac{1}{N}\sum_t (P_y^{(t)})^{-1}\right)^{-1}$	$+0.0026$
Max-pooling ($r \to \infty$)	$\bar{P}_y \propto \max_t P_y^{(t)}$	$+0.0048$

Table 4: Scoring rules organized by family. The DeepConf family uses the per-token top-k confidence $C_i := -\frac{1}{k}\sum_{j=1}^{k} \log p_i^{(j)}$, where $p_i^{(j)}$ denotes the jth largest entry of p_i, and the per-group confidence $C_{G_i} := \frac{1}{w}\sum_{l \in G_i} C_l$ with $k = 20$ and $w = 2048$. $S_{10\%}$ denotes the index set of the bottom 10% groups by C_{G_j}. Vote-based methods do not produce a scalar score; instead, the third column reports the vote weight assigned to each trace or cluster. $\hat{a}(t)$ denotes the decoded answer.

Family	**Method**	**Scoring rule**
Likelihood	Length-normalized log-prob (Wu et al., 2016)	$\phi_{\mathrm{LN}}(t) = \frac{1}{n}\sum_{i=1}^{n} \log p_i(y_i)$
	Perplexity	$\phi_{\mathrm{PPL}}(t) = e^{-\phi_{\mathrm{LN}}(t)}$
Self-confidence	Self-Certainty (Kang et al., 2025)	$\phi_{\mathrm{SC}}(t) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{V} p_i(j)\log\big(V p_i(j)\big)$
	$P(\mathrm{True})$ self-evaluation (Kadavath et al., 2022)	$\phi_{\mathrm{PT}}(t) = p_\theta\big(\mathrm{Yes} \mid q_{\mathrm{verify}}, x, t\big)$
DeepConf (Fu et al., 2026)	Mean confidence	$\phi_{\mathrm{mean}}(t) = \frac{1}{n}\sum_{i=1}^{n} C_i$
	Tail confidence (last w tokens)	$\phi_{\mathrm{tail}}(t) = \frac{1}{w}\sum_{i=n-w+1}^{n} C_i$
	Bottom-10% group confidence	$\phi_{\mathrm{b10}}(t) = \mathrm{mean}_{j \in S_{10\%}} C_{G_j}$
	Lowest group confidence	$\phi_{\mathrm{lgc}}(t) = \min_j C_{G_j}$
Vote-based	Self-Consistency (Wang et al., 2023)	$w(t) \equiv 1$ (uniform vote on $\hat{a}(t)$)
	Soft Self-Consistency (Wang et al., 2024a)	$w(t) = e^{\phi_{\mathrm{LN}}(t)}$
	Self-Certainty + Borda (Kang et al., 2025)	$w(t) = (K - r(t) + 1)^p$, $r(t) := \mathrm{rank}_{\phi_{\mathrm{SC}}}(t)$
	Semantic Entropy (Kuhn et al., 2023; Farquhar et al., 2024)	$w(C_m) = \sum_{t \in C_m} e^{\phi_{\mathrm{LN}}(t)}$, C_m is an NLI cluster

mean ($r = 1$) gives the largest improvement in validation loss, and the geometric mean ($r \to 0$), which is equivalent to logit averaging, is the weakest alternative that outperforms the baseline. The per-coordinate median and the trimmed mean are slightly worse than the arithmetic mean, whereas the harmonic mean and max-pooling are pathological aggregators that amplify the influence of small and large outliers, respectively, causing them to be inferior to even the baseline.

3.3 Reasoning-Trace Selection

For the reasoning tasks, we set (A4) to be a "per-trace estimator," replacing $\bar{P}$ with a scoring rule (Table 4). We also test four aggregation rules $\mathcal{A}$ across the trace pool: BoN (argmax over per-trace score), top-K majority (majority among the top-K traces), `mean_mv2` (per-answer mean of scores among answers with ≥ 2 supporting traces), and `median_mv2` (per-answer median, same filter). For comparison, we also run Verification-Filtered Voting (VFV), in which each trace is independently self-verified $K = 8$ times by the same model and then filtered by verification rate before majority voting. Table 9 in Appendix D lists the detailed results.

Trace separation. Figure 4 demonstrates how SAE is able to solve problems DeepConf that cannot, using BRUMO25 P30 on the 8B model and AIME24 P30 on the 32B model as examples. In the clean log-probability panels, the wrong-trace cluster is readily seen to be heavy-tailed, overlapping with the correct-trace cluster. Furthermore, the mean gap between the clusters is small ($+10.9$ on 8B, only $+3.0$ on 32B), and the top-1 trace is wrong, which causes DeepConf's confidence-based selection to miss the problem. On the other hand, in the SAE panels, the gap between wrong traces and correct traces widens dramatically ($+10.9 \to +21.1$ on 8B, $+3.0 \to +23.2$ on 32B), and the

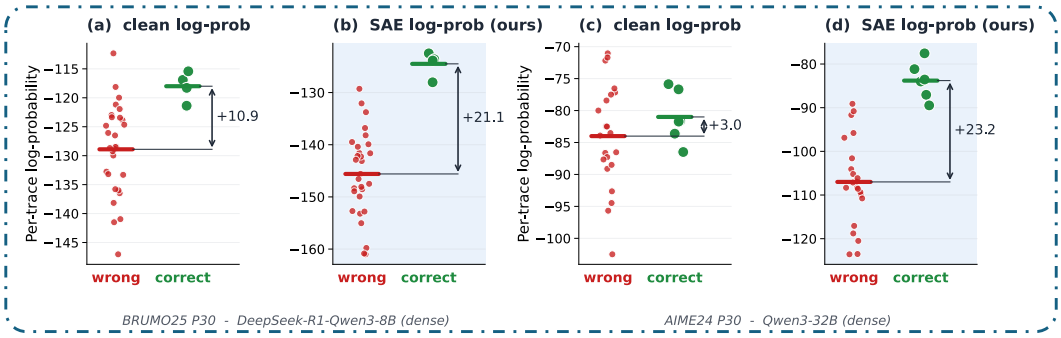

Figure 4: Per-trace log-probability of every reasoning trace from two problems where SAE successfully recovers a correct trace in contrast to the clean-log-prob baseline: BRUMO25 P30 on DeepSeek-R1-0528-Qwen3-8B (panels (a) and (b); 29 wrong traces and 4 correct traces) and AIME24 P30 on Qwen3-32B (panels (c) and (d); 22 wrong traces and 6 correct traces). For each problem, the left panel shows clean inference and the right panel shows SAE with $\sigma = 0.5$.

Table 5: SAE applied to twelve scoring rules from five families. Each method uses its corresponding selection rule (Table 11 in Appendix F); the SAE runs use $N = 16$. Δ denotes the number of problems rescued, with a positive value indicating better performance.

Family	Method	Baseline Acc	SAE (Ours) Acc	Δ
Likelihood	Length-normalized log-prob (Wu et al., 2016)	314/360 (87.2%)	320/360 (88.9%)	+6
	Mean log-prob	318/360 (88.3%)	323/360 (89.7%)	+5
Self-confidence	Self-Certainty BoN (Kang et al., 2025)	310/360 (86.1%)	317/360 (88.1%)	+7
	Self-Certainty + Borda (p^*) (Kang et al., 2025)	314/360 (87.2%)	320/360 (88.9%)	+6
	$P(\text{True})$ self-evaluation (Kadavath et al., 2022)	304/360 (84.4%)	311/360 (86.4%)	+7
DeepConf (Fu et al., 2026)	Mean confidence	312/360 (86.7%)	317/360 (88.1%)	+5
	Tail confidence	313/360 (86.9%)	318/360 (88.3%)	+5
	Bottom-10% group confidence	316/360 (87.8%)	321/360 (89.2%)	+5
	Lowest group confidence	311/360 (86.4%)	316/360 (87.8%)	+5
Vote-based	Soft Self-Consistency (Wang et al., 2024a)	316/360 (87.8%)	321/360 (89.2%)	+5
	Semantic Entropy (Kuhn et al., 2023)	307/360 (85.3%)	313/360 (86.9%)	+6
Lookahead	Lookahead Confidence (Snell et al., 2025)	314/360 (87.2%)	319/360 (88.6%)	+5

correct-trace cluster cleanly rises above the wrong-trace cluster, leading to a correct top-1 trace. This phenomenon is a consistent pattern across model families and benchmarks, enabling a simple BoN over SAE log-probabilities to rescue difficult problems (see Appendix E).

Optimal aggregators. We find the optimal aggregator to be model-dependent (Table 10 in Appendix D). In particular, the 8B model benefits the most from BoN (+3 over DeepConf, rising to +6 when combined with VFV), while the 32B model prefers `median_mv2` (+4). This can be explained by the observation that the 32B model's wrong-trace distributions under SAE are bimodal (e.g. AIME25 P7 has a wrong-trace standard deviation of 40.4, with one outlier outranking 29 correct traces), so the per-answer median is more robust than the per-trace BoN. The MoE again favors BoN (+4).

Robustness of σ. We perform a sweep on a single problem (BRUMO25 P15, 53/251 correct traces, $N = 10$) over $\sigma \in [0.05, 1.0]$ and identify $\sigma \approx 0.5$–0.7 as the regime where the trace separation emerges (Table 12 in Appendix G). Below $\sigma = 0.4$, the noise is too small to perturb predictions, and above $\sigma = 1.0$, the noise overwhelms correct reasoning. We use $\sigma = 0.5$ throughout for reasoning-trace selection, matching the best-performing value for next-token prediction.

Effect of sample size N. Figure 2 shows how the per-trace log-probability distribution changes as we increase the sample size N. The baseline exhibits wrong-trace and correct-trace distributions that heavily overlap, but as N grows from $16 \to 32 \to 64$, the distributions become progressively disjoint and concentrate around their respective means. By $N = 64$, wrong traces are essentially separated from correct traces, so a BoN over the log-probabilities reliably selects a correct trace.

Scores beyond log-probability. SAE's superior reasoning ability is not limited to log-probability scoring. Table 5 reports the result for twelve scoring rules drawn from five families. A positive value of Δ across the board confirms that SAE does not require a particular method—it improves both BoN per-trace scores and vote-based aggregators. Our results suggest that an existing pipeline using Self-Certainty, Self-Certainty + Borda, DeepConf, etc. can readily adopt SAE with no change to the rest of the pipeline.

4 Conclusion

We proposed the TTSE framework, which generalizes existing methods such as MC Dropout, TTA, and stochastic depth and broadly consists of drawing N stochastic forward passes from a single frozen model and aggregating the population to form a final output. We identify a four-axis characterization of TTSE methods and introduce SAE in particular, which adds isotropic Gaussian noise to Query and Key vectors after RoPE and RMSNorm and then averages the probabilities from N passes. Empirically, SAE is consistently superior to the deterministic baseline and stochastic alternatives on next-token prediction and mathematical reasoning tasks.

Future work. The most immediate double-edged sword of TTSE is the deferral of all computational cost to the inference phase. Some natural future directions in this regard include structured-noise instances at small N, low-rank perturbation directions, and amortized aggregators. A second direction is to expand scale and architecture coverage: our next-token-prediction experiments use 300M and 2.7B GPT models on FineWeb with QK-normalization, and our reasoning experiments use 8B–32B reasoning models; evaluating on larger LMs ($\geq$ 7B base, $\geq$ 70B reasoning) and on different architectures would further support the scalability of SAE. Finally, the experiments in this work do not explore the impact of SAE on the quality of generative models or its possible extensions to RLHF/RLVR, which we believe to be perhaps the most exciting direction for future research.

AI use statement

In this work, we used generative AI tools for language editing, including improving grammar, clarity, concision. We have not used generative AI tools to generate experimental results, perform data analysis, or make scientific conclusions, and other required disclosure categories are not applicable to this work. Additionally, we used generative AI tools to suggest alternative wording and improve the readability of selected passages. All AI-assisted text was reviewed and revised by the authors for accuracy, consistency, and appropriate scientific interpretation. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

References

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*, pp. 12375–12396. Association for Computational Linguistics, 2023.

Arsenii Ashukha, Andrei Atanov, and Dmitry P. Vetrov. Mean embeddings with test-time data augmentation for ensembling of representations. *CoRR*, abs/2106.08038, 2021.

Mirza Samad Ahmed Baig, Syeda Anshrah Gillani, Abdul Akbar Khan, and Shahid Munir Shah. AttentionDrop: A novel regularization method for Transformer models. *CoRR*, abs/2504.12088, 2025.

Brown University Mathematical Olympiad. BRUMO 2025 problems. `https://www.brumo.org/archive`, 2025.

Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*, pp. 295–305, 2022.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. Universal self-consistency for large language model generation. *CoRR*, abs/2311.17311, 2023.

B. V. Dasarathy and B. V. Sheela. A composite classifier system design: Concepts and methodology. *Proceedings of the IEEE*, 67(5):708–713, 1979.

DeepSeek-AI et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *CoRR*, abs/2501.12948, 2025.

Mario Döbler, Robert A. Marsden, and Bin Yang. Robust mean teacher for continual and gradual test-time adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023*, pp. 7704–7714, 2023.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *The Tenth International Conference on Learning Representations, ICLR 2022*, 2022.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *CoRR*, abs/1912.02757, 2019.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3259–3269, 2020.

Yichao Fu, Xuewei Wang, Hao Zhang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence. In *The Fourteenth International Conference on Learning Representations, ICLR 2026*, 2026.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 1050–1059, 2016.

Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei A. Efros. Test-time training with masked autoencoders. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, pp. 29374–29385, 2022.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P. Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of DNNs. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, pp. 8803–8812, 2018.

Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. NOTE: robust continual test-time adaptation against temporal correlation. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, 2022.

Taesik Gong, Yewon Kim, Taeckyung Lee, Sorn Chottananurak, and Sung-Ju Lee. SoTTA: Robust test-time adaptation on noisy data streams. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*, 2023.

Jisu Han, Jaemin Na, and Wonjun Hwang. Ranked entropy minimization for continual test-time adaptation. In *Forty-second International Conference on Machine Learning, ICML 2025*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21798–21813, 2025.

Harvard-MIT Mathematics Tournament. HMMT February 2025 problems. https://www.hmmt.org/www/archive/282, 2025.

Jinwu Hu, Zitian Zhang, Guohao Chen, Xutao Wen, Chao Shuai, Wei Luo, Bin Xiao, Yuanqing Li, and Mingkui Tan. Test-time learning for large language models. In *Forty-second International Conference on Machine Learning, ICML 2025*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24823–24849, 2025.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q. Weinberger. Deep networks with stochastic depth. In *Computer Vision - ECCV 2016 - 14th European Conference, Proceedings, Part IV*, volume 9908 of *Lecture Notes in Computer Science*, pp. 646–661. Springer, 2016.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018*, pp. 876–885, 2018.

Minguk Jang, Sae-Young Chung, and Hye Won Chung. Test-time adaptation via self-training with nearest neighbor information. In *The Eleventh International Conference on Learning Representations, ICLR 2023*, 2023.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know. *CoRR*, abs/2207.05221, 2022.

Zhewei Kang, Xuandong Zhao, and Dawn Song. Scalable best-of-N selection for large language models via self-certainty. In *Advances in Neural Information Processing Systems 39: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025*, 2025.

Byungjai Kim, Chanho Ahn, Wissam J. Baddar, Kikyung Kim, Huijin Lee, Saehyun Ahn, Seungju Han, Sungjoo Suh, and Eunho Yang. Test-time ensemble via linear mode connectivity: A path to better adaptation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025*, 2025.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations, ICLR 2023*, 2023.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pp. 6402–6413, 2017.

Jonghyun Lee, Dahuin Jung, Saehyung Lee, Junsung Park, Juhyeon Shin, Uiwon Hwang, and Sungroh Yoon. Entropy is not enough for test-time adaptation: From the perspective of disentangled factors. In *The Twelfth International Conference on Learning Representations, ICLR 2024*, 2024.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024*, 2024.

Hyesu Lim, Byeonggeun Kim, Jaegul Choo, and Sungha Choi. TTN: A domain-shift aware batch normalization in test-time adaptation. In *The Eleventh International Conference on Learning Representations, ICLR 2023*, 2023.

Wesley J. Maddox, Pavel Izmailov, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. A simple baseline for Bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, pp. 13132–13143, 2019.

Mathematical Association of America. AIME 2024 problems. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions, 2024.

Mathematical Association of America. AIME 2025 problems. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions, 2025.

Saypraseuth Mounsaveng, Florent Chiaroni, Malik Boudiaf, Marco Pedersoli, and Ismail Ben Ayed. Bag of tricks for fully test-time adaptation. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2024*, pp. 1925–1934, 2024.

Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, 2020.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International Conference on Machine Learning, ICML 2022*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16888–16905, 2022.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *The Eleventh International Conference on Learning Representations, ICLR 2023*, 2023.

Shuaicheng Niu, Chunyan Miao, Guohao Chen, Pengcheng Wu, and Peilin Zhao. Test-time model adaptation with only forward passes. In *Forty-first International Conference on Machine Learning, ICML 2024*, Proceedings of Machine Learning Research, pp. 38298–38315, 2024.

Jiahuan Pei, Cheng Wang, and György Szarvas. Transformer uncertainty estimation with hierarchical stochastic attention. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022*, pp. 11147–11155, 2022.

Guilherme Penedo, Hynek Kydlícek, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin A. Raffel, Leandro von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Datasets and Benchmarks Track*, 2024.

Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025.

Jason Ramapuram, Federico Danieli, Eeshan Gunesh Dhekane, Floris Weers, Dan Busbridge, Pierre Ablin, Tatiana Likhomanenko, Jagrit Digani, Zijin Gu, Amitis Shidani, and Russell Webb. Theory, analysis, and best practices for sigmoid self-attention. In *The Thirteenth International Conference on Learning Representations, ICLR 2025*, 2025.

Alexandre Ramé, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. Diverse weight averaging for out-of-distribution generalization. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, 2022.

R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. FlashAttention-3: Fast and accurate attention with asynchrony and low-precision. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024*, 2024.

Divya Shanmugam, Davis W. Blalock, Guha Balakrishnan, and John V. Guttag. Better aggregation in test-time augmentation. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021*, pp. 1194–1203, 2021.

Noam Shazeer. GLU variants improve Transformer. *CoRR*, abs/2002.05202, 2020.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025*, 2025.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9229–9248, 2020.

Mingkui Tan, Guohao Chen, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Peilin Zhao, and Shuaicheng Niu. Uncertainty-calibrated test-time model adaptation without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(8):6274–6289, 2025.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. Confidence improves self-consistency in LLMs. In *Findings of the Association for Computational Linguistics, ACL 2025*, Findings of ACL, pp. 20090–20111, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pp. 5998–6008, 2017.

Akshay Vegesna, Samip Dahal, Chinmay Karkar, Bishwas Mandal, Shmuel Berman, and Zhiwei Xu. Slowrun: Language modeling with infinite compute, fixed data. `https://github.com/qlabs-eng/slowrun`, 2026.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno A. Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *9th International Conference on Learning Representations, ICLR 2021*, 2021.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Soft self-consistency improves language model agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2024*, pp. 287–301, 2024a.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024*, pp. 9426–9439, 2024b.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*, pp. 7191–7201, 2022.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023*, 2023.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning, ICML 2022*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23965–23998, 2022a.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*, pp. 7949–7961, 2022b.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.

Longhui Yuan, Binhui Xie, and Shuang Li. Robust test-time adaptation in dynamic scenarios. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023*, pp. 15922–15932, 2023.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, pp. 12360–12371, 2019.

Jianyu Zhang and Léon Bottou. Fine-tuning with very large dropout. *CoRR*, abs/2403.00946, 2024.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction. In *The Thirteenth International Conference on Learning Representations, ICLR 2025*, 2025a.

Marvin Zhang, Sergey Levine, and Chelsea Finn. MEMO: test time robustness via adaptation and augmentation. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, 2022.

Qingyang Zhang, Yatao Bian, Xinke Kong, Peilin Zhao, and Changqing Zhang. COME: test-time adaption by conservatively minimizing entropy. In *The Thirteenth International Conference on Learning Representations, ICLR 2025*, 2025b.

Bowen Zhao, Chen Chen, and Shu-Tao Xia. DELTA: Degradation-free fully test-time adaptation. In *The Eleventh International Conference on Learning Representations, ICLR 2023*, 2023.

A Related Work

No prior work presents TTSE as a unified framework along the four axes of Section 2.2; nevertheless, we discuss several closely related methods in this section.

Training-time ensembles incorporate the ensemble structure into θ prior to inference (Dasarathy & Sheela, 1979; Schapire, 1990; Lakshminarayanan et al., 2017; Fort et al., 2019), including methods that perturb training itself (Huang et al., 2016; Zhang & Bottou, 2024; Pei et al., 2022; Maddox et al., 2019; Izmailov et al., 2018; Garipov et al., 2018) or average across linearly-connected modes (Frankle et al., 2020; Neyshabur et al., 2020; Entezari et al., 2022; Wortsman et al., 2022a;b; Ramé et al., 2022). An alternative approach instead updates θ at inference to handle distribution shift via entropy minimization (Wang et al., 2021; Niu et al., 2022; 2023; Lee et al., 2024; Zhang et al., 2025b; Han et al., 2025; Tan et al., 2025; Niu et al., 2024), self-supervised test-time training (Sun et al., 2020; Gandelsman et al., 2022; Zhang et al., 2022), continual or streaming variants (Wang et al., 2022; Gong et al., 2022; 2023; Yuan et al., 2023; Döbler et al., 2023; Chen et al., 2022; Lim et al., 2023; Zhao et al., 2023; Mounsaveng et al., 2024; Jang et al., 2023), language-model extensions (Hu et al., 2025), or weight-space test-time ensembling (Kim et al., 2025). Both regimes retrain θ or require an unsupervised adaptation loss; in contrast, TTSE holds θ frozen and obtains diversity entirely from a noise process at inference.

Methods closest to TTSE each fix one or two of the design axes. MC Dropout (Gal & Ghahramani, 2016) fixes (A1) to activations and (A2) to Bernoulli, which produces little diversity on modern low-dropout transformers (Table 2); test-time augmentation (Shanmugam et al., 2021; Ashukha et al., 2021) fixes (A1) to the input; AttentionDrop (Baig et al., 2025) fixes (A1) to attention logits with random masking or Gaussian blurring, but only as a training-time regularizer; and sigmoid attention (Ramapuram et al., 2025) is a deterministic architectural alternative orthogonal to TTSE. For reasoning-trace selection, self-consistency (Wang et al., 2023; Chen et al., 2023) fixes (A4) to a discrete vote over N decoded answers and discards the per-pass distributions, a setting since studied systematically (Snell et al., 2025) and refined into soft (Wang et al., 2024a), confidence-improved (Taubenfeld et al., 2025), and adaptive (Aggarwal et al., 2023) consistency. Candidate trace-level scores include self-certainty (Kang et al., 2025), length-normalized log-probability (Wu et al., 2016), semantic uncertainty (Kuhn et al., 2023; Farquhar et al., 2024), internal-knowledge probes (Kadavath et al., 2022), process-level verifiers (Wang et al., 2024b; Lightman et al., 2024), generative verifiers (Zhang et al., 2025a), and DeepConf (Fu et al., 2026), our direct baseline. SAE composes with any of these scores on a frozen reasoning LM, using the per-pass population to sharpen the correct-vs-wrong gap (Fig. 2).

B Noise Scale and Sample Size Sweep

Table 6: SAE noise scale and sample size sweep.

	Sample size N					
σ	5	10	20	30	50	100
0.1	−0.0005	−0.0008	−0.0010	−0.0011	−0.0012	−0.0013
0.3	−0.0038	−0.0052	−0.0060	−0.0064	−0.0070	−0.0073
0.5	−0.0070	−0.0097	−0.0118	−0.0123	−0.0134	−0.0140
0.6	−0.0059	−0.0086	−0.0107	−0.0112	−0.0120	−0.0124
0.8	+0.0143	+0.0081	+0.0046	+0.0035	+0.0022	+0.0015

Table 7: Probability vs. logit averaging across noise types ($N = 50$, $\sigma = 0.5$).

Noise Type	Logit Averaging Δ	Probability Averaging Δ	Ratio
Q + K noise ($\sigma = 0.5$)	-0.0070	-0.0134	$1.9\times$
Attention logits (Gumbel)	-0.0063	-0.0123	$2.0\times$
Weight perturbation	-0.0004	-0.0009	$2.3\times$

Table 8: Effect of averaging strategy on a 14-model 2.7B GPT ensemble. Switching from logit averaging to probability averaging is a free improvement at zero additional compute.

Configuration	Val Loss	Δ vs. Logit Averaging
Logit averaging (standard)	3.0919	—
Probability averaging (ours)	3.0824	-0.0095
Probability averaging + noise ($\sigma = 0.3$, $N = 3$)	3.0827	-0.0092

C Probability vs. Logit Averaging

D Reasoning Results

Table 9: Reasoning accuracy across 360 problems (3 models $\times$ 4 datasets $\times$ 30 problems). All SAE/VFV methods applied post hoc to DeepConf's online traces.

Method	Score	Accuracy	Δ
DeepConf (best of 7 confidence variants) (baseline)	316/360	87.8%	—
SAE BoN	323/360	89.7%	+7
SAE top-3 majority	322/360	89.4%	+6
SAE top-5 majority	322/360	89.4%	+6
SAE mean_mv2	323/360	89.7%	+7
SAE median_mv2	323/360	89.7%	+7
Smart (agreement-based)	322/360	89.4%	+6
Weighted ensemble	320/360	88.9%	+4
raw_majority (all traces)	318/360	88.3%	+2
Consensus-ensemble (8-method vote)	**325/360**	**90.3%**	+9
Oracle (any SAE method correct)	328/360	91.1%	+12
Oracle (any SAE+VFV method correct)	330/360	91.7%	+14

Table 10: Per-model breakdown (4 datasets, 30 problems each). *Best aggregator* is the one that produces the per-dataset gain.

Model	Dataset	DC	Best SAE	Δ	Best aggregator
8B dense	BRUMO25	28	29	+1	BoN
	AIME24	28	28	0	—
	AIME25	26	26	0	—
	HMMT25	24	26	+2	BoN
	Total	*106*	*109*	+3	+ VFV → 112 (+6)
32B dense	BRUMO25	28	29	+1	BoN
	AIME24	26	27	+1	top-5
	AIME25	25	27	+2	median_mv2
	HMMT25	21	22	+1	median_mv2
	Total	*100*	*104*	+4	bimodal wrong traces
30B-A3B MoE	BRUMO25	∼24	∼24	0	—
	AIME24	28	28	0	0 correct in pool
	AIME25	26	28	+2	BoN
	HMMT25	24	26	+2	BoN
	Total	∼102	∼106	+4	complementary to DC top-10%

E Per-Problem Breakdown for Reasoning Verification

This appendix lists every DeepConf failure across the 360 problems and which method (SAE / VFV / neither) rescues it. "Correct/Valid" is the count of traces in the pool with the correct answer over the total valid traces.

E.1 DeepSeek-R1-0528-Qwen3-8B

BRUMO25 (P30 rescued)

Problem	GT	Correct/Valid	SAE	VFV	Mechanism
P23	complex roots	0/78	✗	✗	impossible (0 correct)
P30	$9\sqrt{30}/4$	4/33 (12%)	✓	✗	SAE BoN (gap +21.1)

AIME24 (P26 rescued)

Problem	GT	Correct/Valid	SAE	VFV	Mechanism
P4	385	0/53	✗	✗	impossible
P26	080	14/36 (39%)	✗	✓	VFV (threshold $\geq 25\%$)

AIME25 (P13 rescued)

Problem	GT	Correct/Valid	SAE	VFV	Mechanism
P13	204	4/34 (12%)	✗	✓	VFV (verify rate 75%)
P14	60	0/39	✗	✗	impossible
P15	735	1/45 (2%)	✗	✗	too few correct
P30	240	3/47 (6%)	✗	✗	model self-verifies wrong "188" at 100%

HMMT25 (P13, P18, P24 rescued)

Problem	GT	Correct/Valid	SAE	VFV	Mechanism
P13	105	3/71 (4%)	✗	✓	VFV (threshold $\geq 25\%$)
P17	$2^{25} \cdot 26!$	0/103	✗	✗	impossible
P18	$2025/101$	9/41 (22%)	✓	✗	SAE BoN (gap +18.6)
P24	20	1/27 (4%)	✓	✓	SAE BoN + VFV

E.2 QWEN3-32B DENSE

BRUMO25 (P30 rescued)

Problem	GT	Correct/Valid	SAE	Mechanism
P23	complex roots	2/228	✗	impossible
P30	$9\sqrt{30}/4$	2/67 (3%)	✓	SAE BoN

AIME24 (P30 rescued)

Problem	GT	Correct/Valid	SAE	Mechanism
P3	371	10/70 (14%)	✗	SAE gap correct but BoN outlier
P4	385	0/159	✗	impossible
P22	315	0/119	✗	impossible
P30	902	31/118 (26%)	✓(top-5)	4 of top-5 SAE traces correct

AIME25 (P14 + P28 rescued)

Problem	GT	Correct/Valid	BoN	median_mv2	Mechanism
P7	821	29/103 (28%)	✗	✗	gap $+6.2$ but BoN picks outlier "271"
P13	204	2/46 (4%)	✗	✗	gap $+14.4$ but rank 11
P14	60	2/38 (5%)	✗	✓	median_mv2 (2 consistent traces)
P15	735	0/60	✗	✗	impossible
P28	248	4/37 (11%)	✗	✓	median_mv2

HMMT25 (P15 rescued)

Problem	GT	Correct/Valid	SAE	Mechanism
P10/P14/P19/P20	various	0 correct each	✗	impossible (4 problems)
P12	29	9/68 (13%)	✗	SAE fails
P13	105	2/93 (2%)	✗	SAE fails
P15	200	10/111 (9%)	✓(median_mv2)	answer-level median rescues
P17	$2^{25}\cdot 26!$	1/119	✗	only 1 correct trace
P18	$2025/101$	12/81 (15%)	✗	wrong outlier "675/67" dominates
P24	20	19/120 (16%)	✗	SAE fails (wrong "28" dominant)

E.3 QWEN3-30B-A3B MOE

AIME25 (P15 + P29 rescued)

Problem	GT	Correct/Valid	SAE (after fix)	Mechanism
P13	204	12/47 (26%)	✗	SAE gap -0.7 (too close)
P14	60	0/1	✗	impossible
P15	735	5/46 (11%)	✓	SAE BoN (gap $+7.2$)
P29	104	10/35 (29%)	✓	SAE BoN

HMMT25 (P11 + P29 rescued; P17 + P18 lost to DeepConf top-10%)

Problem	GT	Correct/Valid	SAE	DeepConf	Winner
P11	56	35/57 (61%)	✓	✗ (filter empties)	SAE
P13	105	9/42 (21%)	✗	✗	neither
P17	$2^{25}\cdot 26!$	10/46	✗	✓(top-10%)	DC
P18	$2025/101$	20/66	✗	✓(top-10%)	DC
P29	$14+4\sqrt{37}$	2/5	✓	✗	SAE
P23/P24	various	various	✗	✗	neither

F Corresponding Selection Rules

Table 11: The corresponding selection rule for each scoring rule. "BoN" means top-1 trace by the base score; vote-based methods use their published aggregator. The same rule is applied to both the baseline and SAE columns of Table 5.

Selection rule	**Scoring rules**
BoN (top-1 by score)	Length-norm log-prob, Mean log-prob, Self-Certainty BoN, $P(\text{True})$, DeepConf Mean / Tail / Bottom-10% / Lowest-Group, Lookahead Confidence
BoN (lowest entropy)	Semantic Entropy
Borda count vote	Self-Certainty + Borda
Confidence-weighted majority	Soft Self-Consistency

G Reasoning Noise Scale Sweep

Table 12: Sweep on BRUMO25 P15 ($N = 10$, 53/251 correct traces). Below $\sigma = 0.4$ the noise is too small to perturb predictions and SAE scores match clean log-probabilities; at $\sigma = 1.0$ the noise overwhelms correct reasoning and the gap inverts.

σ	0.05	0.1	0.2	0.3	0.4	0.5	0.7	1.0
Correct trace SAE	-0.0	-0.0	-0.0	-0.0	-0.0	-0.1	-1.3	-28.4
Wrong trace SAE	-0.0	-0.0	-0.0	-0.0	-0.0	-0.1	-1.8	-26.3
Gap (correct$-$wrong)	0	0	0	0	0	weak	$+0.5$	-2.1

H MoE Attention-Class Bug

We discovered a portability-relevant bug specific to mixture-of-experts deployments. Initial MoE results showed *zero* SAE signal: per-trace SAE means equaled the clean log-probabilities and per-trace standard deviations across N samples were exactly 0.

The root cause was due to our monkey-patch targeted `Qwen3Attention.forward` (from `transformers.models.qwen3`).

The MoE model uses `Qwen3MoeAttention` (from `transformers.models.qwen3_moe`), which is a separate class object even though the forward source is byte-for-byte identical:

```
inspect.getsource(Qwen3Attention.forward) ==
    inspect.getsource(Qwen3MoeAttention.forward)    # True
Qwen3Attention is Qwen3MoeAttention                 # False
```

The fix is to patch both classes. After the fix, MoE AIME25 went from $+0$ to $+2$ rescues (P15, P29). Any noise-injection scheme via Python class patching must enumerate every architecture's attention class explicitly.