WATERMARKING CONTINUOUS DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing watermarking methods for text generation focus on autoregressive and discrete diffusion language models, leaving continuous diffusion language models (CDLMs) unexplored. To close this gap, we introduce **COMPASS** (**C**ontinuous-Latent **O**utput **M**arking for **P**rovenance via **A**nchor **S**teering and **S**coring), a training-free watermark designed specifically for CDLMs. COMPASS constructs key-dependent target and non-target anchor representations in the latent diffusion space and, during a selected denoising window, steers latent vectors associated with non-target anchors toward the target anchor set. Detection requires only the candidate text. The detector re-encodes the text using the CDLM's encoder and measures whether the resulting latent representations align with the target anchors defined by the watermark key. We show that this design provides robust detection of content generated by state-of-the-art CDLMs, namely Cola and ELF-L, while preserving their high-quality outputs. Furthermore, COMPASS requires neither model fine-tuning nor additional neural-network evaluations during generation, providing an efficient and practical framework for provenance tracing in CDLMs.

1 INTRODUCTION

Modern large language models (LLMs) are mainly based on autoregressive generation, in which text is produced one token at a time in a fixed left-to-right order (Vaswani et al., 2017; Touvron et al., 2023). While this paradigm has led to significant progress in language modeling, its sequential decoding process does not naturally support any-order generation or revision of previously generated tokens, and it limits parallel decoding, which together can limit generation flexibility and efficiency.

These limitations have motivated alternative generation paradigms, such as diffusion language models (DLMs), including *discrete* diffusion language models (DDLMs) (Nie et al., 2025; Sahoo et al., 2024; Lou et al., 2024; Austin et al., 2021) and *continuous* diffusion language models (CDLMs) (Li et al., 2022; Hu et al., 2026; Jo & Hwang, 2025; Guo et al., 2026), which, instead of generating tokens sequentially, iteratively refine multiple token representations through a diffusion process.

The emergence of these architectures also creates a need to distinguish their outputs from human-written text, both to identify synthetic content and to establish data provenance. Watermarking offers a practical approach: it embeds a secret signal into generated content that can later be detected (Kirchenbauer et al., 2023; Zhao et al., 2024). Most existing language watermarks are designed for autoregressive models (Liang et al., 2025), where a common strategy divides the vocabulary into key-dependent groups and increases the probability of selected tokens during generation. When the next-token distribution has sufficient entropy, these small probability shifts can accumulate into a detectable signal without substantially altering the generated text.

However, these autoregressive mechanisms do not transfer naturally to diffusion models, and prior work has therefore introduced watermarking schemes designed specifically for DDLMs (Chen et al., 2025; Gloaguen et al., 2026; Hong & No, 2026). These methods nevertheless operate on *discrete* generation decisions: they either bias token selection or control the order in which positions are unmasked. Neither mechanism has a direct counterpart in CDLMs, whose denoising process evolves in *continuous representations* instead of discrete tokens and mask states. As recent work (Zhou et al., 2026) suggests that continuous diffusion models can be more expressive than discrete diffusion models and looped transformers (Giannou et al., 2023), the absence of watermarking methods tailored specifically to this setting leaves an important gap.

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.

Adapting token-level watermarks to CDLMs is challenging. Although these models ultimately produce discrete tokens, generation happens in a continuous latent space, and we find their output token distributions can be extremely concentrated, with the maximum probability often close to one (Appx. A). In this regime, small changes to token probabilities produce little detectable signal, whereas larger changes are more likely to alter the selected token and degrade the generated text. Token-level watermarking therefore faces an unfavorable trade-off between detection and generation quality.

To address this challenge, we introduce COMPASS (**C**ontinuous-Latent **O**utput **M**arking for **P**rovenance via **A**nchor **S**teering and **S**coring), a training-free watermark designed for CDLMs. Instead of modifying token probabilities or the initial diffusion noise, COMPASS embeds the watermark directly into the continuous denoising trajectory. It constructs a key-dependent set of anchor representations in the model's latent space and partitions them into target and non-target groups. During selected denoising steps, representations assigned to non-target regions are steered toward randomly selected target anchors. The latents do not need to stay at the anchors. The remaining denoising steps refine them, but a measurable bias toward the target anchors survives. The detector recovers this bias by re-encoding the candidate text with the model's encoder and measuring how its representations align with the anchors derived from the secret key.

We evaluate COMPASS on two state-of-the-art CDLMs, Cola (Guo et al., 2026) and ELF-L (Hu et al., 2026), against eight watermarking baselines adapted from autoregressive text generation and image diffusion. On Cola, COMPASS achieves a 94.2% true-positive rate at a 1% false-positive rate, compared with 89.2% for the strongest baseline, while incurring the lowest judged quality degradation among the evaluated methods (9.1%), and on ELF-L, it achieves a 100.0% true-positive rate with 4.3% degradation. The advantage persists after post-generation editing: across both models, COMPASS remains substantially more robust than the evaluated baselines to a wide range of modifications, including deletion, surface edits, clean-text dilution, and paraphrasing.

In summary, our contributions are as follows:

1. We introduce COMPASS, the first watermark designed specifically for CDLMs. It embeds a key-dependent directional signal during denoising, without retraining or adding forward passes.
2. We adapt eight baseline watermarks from autoregressive models and image diffusion to CDLMs and show that token-level methods fail because CDLM output distributions are nearly one-hot.
3. We show that the probability of a steered latent remaining in a target anchor region predicts detection success, motivating how codebook size and watermarking window should be chosen.
4. We evaluate COMPASS against eight baselines, demonstrating superior detection and robustness to post-generation edits while maintaining high text quality.

2 Background and Related Work

Large language models. Autoregressive language models (Touvron et al., 2023; Grattafiori et al., 2024) sample one token at a time, conditioning each prediction on the tokens already generated. Diffusion language models (Austin et al., 2021; Lou et al., 2024) instead refine a block over multiple steps. In masked discrete diffusion, the state is a partially specified token sequence, and the sampler decides which masked positions to reveal (Sahoo et al., 2024; Nie et al., 2025). One step predicts every masked position at once, and how many positions are actually revealed per step is a sampling choice. Continuous diffusion language models (CDLMs), *e.g.,* Cola (Guo et al., 2026) and ELF-L (Hu et al., 2026) studied in this work, denoise real-valued states, such as token embeddings or learned autoencoder latents, and convert them to text after denoising (Li et al., 2022; Lovelace et al., 2023).

Watermarking autoregressive models. Watermarking embeds a signal in generated data so that a detector can later recognize that data as model output. The most common watermarking approaches for autoregressive models build on the red-green-list method (Kirchenbauer et al., 2023). This watermarking method uses a key and the preceding tokens to select a preferred subset of the vocabulary. It raises the logits of preferred tokens during generation and detects this bias in the output. Distribution-preserving methods take a different route, coupling token sampling to keyed randomness while leaving the model's output distribution unchanged when averaged over the key (Kuditipudi et al., 2024). Both approaches rely on discrete choices among plausible next tokens, and are therefore not directly applicable to CDLMs as we analyze in Appx. A.

Watermarking discrete diffusion models. The left-context hash used by an autoregressive watermark is not a natural choice when tokens can be filled in any order, as done in discrete diffusion models. This is because some of the tokens required to compute the hash may still be masked. Gloaguen et al. (2026) handle this by averaging the watermark bias over uncertain context tokens and favoring choices that strengthen the watermark at other positions. dgMARK (Hong & No, 2026) uses a separate degree of freedom, the order of unmasking, to favor watermark-compatible token predictions. These methods address the missing-prefix problem without imposing left-to-right generation. Their mechanisms depend, however, on a *discrete* state with masked and revealed positions. CDLMs, in contrast, refine *continuous* vectors, so the same mechanisms do not transfer.

Sentence-level semantic watermarks. SemStamp (Hou et al., 2024) hashes sentence embeddings into keyed regions and rejects candidates outside them, tying the watermark to sentence meaning and improving robustness to paraphrasing. The cost is repeated generation: for locality-sensitive hashing with dimension $d = 3$ and rejection margin $m = 0.02$, the authors report that SemStamp is $20.9\times$ slower than unwatermarked generation. Applying the same rejection mechanism to a CDLM would require a fresh denoising run for every rejected candidate. COMPASS instead embeds its signal within the original denoising trajectory and therefore avoids this repeated-generation overhead (Appx. I).

Watermarking diffusion trajectories. In image diffusion, Tree-Ring (Wen et al., 2023) embeds a Fourier pattern in the initial noise, and Gaussian Shading (Yang et al., 2024) encodes a message in noise drawn from a prescribed distribution. Both recover the signal by inverting the image-generation process. We adapt them as continuous-latent baselines in Sec. 4. Yet, applying watermarks to text instead of images poses an additional bottleneck: the final continuous state is decoded into discrete tokens, so a signal in the initial noise must remain recoverable from those tokens. Because decoding is many-to-one and the detector observes only the resulting text, re-encoding cannot recover the exact latent state that carried the watermark. Inversion therefore begins from a different state, weakening recovery of the embedded signal. This limitation is reflected in our experiments, where the adapted image-diffusion baselines provide weaker and less robust detection than COMPASS. Guidance Watermarking (Gesny et al., 2026) intervenes during image denoising using gradients from a differentiable watermark detector. COMPASS also intervenes during denoising but steers representations directly using keyed anchor directions. It therefore avoids the additional detector evaluations and backpropagation required by gradient-based guidance.

3 Our Watermark for CDLMs

COMPASS places a key-dependent directional bias in intermediate latents during denoising and tests for that bias in representations recovered from candidate text.

3.1 Overview of our Watermark

Generation unit. CDLMs generate text by iteratively refining continuous latent representations. We call the collection of latent positions refined jointly a *generation unit*. Its size remains fixed throughout denoising. COMPASS operates on one generation unit at a time during selected denoising steps, leaving previously generated context, conditioning inputs, and model parameters unchanged.

Watermark injection. Using a secret key, COMPASS samples anchor vectors from a fixed model-aligned reference distribution, normalizes them to unit length, and partitions them evenly into target and non-target sets. At each step within the predefined watermarking window, COMPASS identifies latent vectors whose nearest anchors are non-target and selects a predefined fraction for watermarking. Each selected direction is steered toward a randomly sampled target anchor using spherical linear interpolation (SLERP) (Shoemake, 1985). A small diversity perturbation is then added, and the latent's original norm is restored. Latents assigned to target anchors and non-target latents not selected for watermarking remain unchanged. The resulting state is passed to the next denoising step, and this procedure is repeated while the step remains within the watermarking window. Once the window ends, standard denoising continues without further modification. Figure 1 illustrates the watermarking window and the subsequent denoising process.

Detection. Given a candidate text, the detector uses the model's frozen encoder to recover latent representations and measures their preference for the keyed target anchors over the non-target anchors.

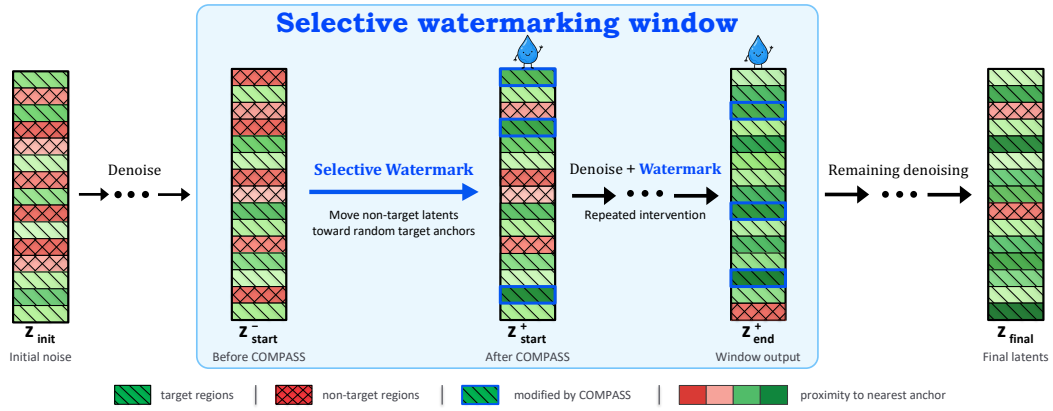


Figure 1: **Overview of COMPASS.** During a predefined denoising window, a selected fraction of the latent vectors that fall within key-defined non-target regions are moved toward randomly sampled target anchors while preserving their norms. Vectors already assigned to target regions remain unchanged. After watermark injection, standard denoising resumes and the final latent representations are decoded into text. Labels denote the initial noise ($\mathbf{z}_{\text{init}}$), the latents at the start of the window before ($\mathbf{z}^{-}_{\text{start}}$) and after ($\mathbf{z}^{+}_{\text{start}}$) COMPASS, the window output ($\mathbf{z}^{+}_{\text{end}}$), and the final latents ($\mathbf{z}_{\text{final}}$).

Detection therefore does not require access to the latents recorded during generation. Full injection and detection algorithms are given in Appx. B.

The position of the watermarking window controls the trade-off between detectability and generation quality. When watermarking occurs during the earliest, noise-dominated steps, the resulting signal is substantially weakened by subsequent denoising. Our analysis shows that, particularly for larger codebooks, the final occupancy of the target regions approaches chance when watermarking is applied too early (Figure 8; see Appx. E for the derivation). Applying the watermark near the end of denoising preserves the keyed signal more strongly but leaves the model fewer steps in which to reconcile the modified latents with their surrounding context, resulting in greater quality degradation. We therefore use an intermediate watermarking window: late enough for the directional bias to survive denoising and the subsequent decoding and re-encoding of the generated text, but early enough for the remaining unmodified steps to restore contextual consistency.

3.2 Latent Representations in CDLMs

COMPASS requires access only to the intermediate latents of the current *generation unit*. At denoising step s, let

$$Z^{(s)} = \left[z_1^{(s)}, \ldots, z_L^{(s)}\right]^\top \in \mathbb{R}^{L \times d}$$

denote the latent state immediately before COMPASS is applied at that step, where L is the fixed number of latent positions and d is the latent dimension. Previously generated context and other conditioning inputs remain unchanged.

Let $\mathcal{S}_{\text{wm}}$ denote the predefined set of active watermarking steps. For $s \in \mathcal{S}_{\text{wm}}$, COMPASS modifies a selected fraction of the latents whose nearest anchors are non-target while leaving all remaining latent positions unchanged. We denote the resulting watermarked state by

$$\widetilde{Z}^{(s)} = \mathcal{W}_{\kappa,s}\left(Z^{(s)}\right),$$

where $\mathcal{W}_{\kappa,s}$ is the keyed watermarking transformation described in Sec. 3.4.

The diffusion trajectory then evolves according to

$$Z^{(s+1)} = \begin{cases} \Phi_s\left(\widetilde{Z}^{(s)}\right), & s \in \mathcal{S}_{\text{wm}}, \\ \Phi_s\left(Z^{(s)}\right), & s \notin \mathcal{S}_{\text{wm}}, \end{cases}$$

where Φ_s denotes the transformation performed by the diffusion backbone from step s to step $s+1$.

3.3 Construction of Keyed Anchor Representations

COMPASS derives a codebook of K unit-norm anchors a_k ($k \in \{1, \ldots, K\}$) from a secret key κ and a fixed model-aligned reference distribution:

$$\mathcal{A}_\kappa = \{(a_k, y_k)\}_{k=1}^{K}, \qquad a_k \in \mathbb{R}^d, \qquad \|a_k\|_2 = 1,$$

where the label $y_k \in \{T, N\}$ marks each anchor as target or non-target.

Anchors are only useful in regions where the model's latents actually lie. If the anchor directions ignore the geometry of the latent space, most latents fall into the cells of a few anchors, and steering pushes them toward directions the model rarely produces. We therefore draw anchors from a reference distribution $\mathcal{D}_M$ that reflects the directions of model M's latents. $\mathcal{D}_M$ depends only on the model, not on the key or the evaluation texts, so the generator and the detector share it. Since latent geometry differs across models, $\mathcal{D}_M$ is model specific (Appx. C).

The key κ is an integer seed for a deterministic pseudorandom generator G_κ, which samples K candidates and normalizes them:

$$(b_1, \ldots, b_K) = \mathrm{Sample}(G_\kappa, \mathcal{D}_M, K)\,, \qquad a_k = \frac{b_k}{\|b_k\|_2}.$$

G_κ then splits the anchors into $K/2$ target anchors $\mathcal{A}_\kappa^T$ and $K/2$ non-target anchors $\mathcal{A}_\kappa^N$, with K even. The equal split ensures that neither set is favored simply by having more anchors. Since the key determines both the directions and the split, the generator and the detector reconstruct the same codebook from κ and $\mathcal{D}_M$, and the codebook never needs to be stored.

To evaluate how a given latent representation aligns with these subsets, we consider a nonzero latent z, and compare its direction

$$u(z) = \frac{z}{\|z\|_2} \tag{1}$$

with the two anchor sets using cosine similarity. We define the keyed anchor margin as

$$\Delta_\kappa(z) = \max_{a \in \mathcal{A}_\kappa^T} \langle u(z), a \rangle - \max_{a \in \mathcal{A}_\kappa^N} \langle u(z), a \rangle. \tag{2}$$

A latent lies in a target region when $\Delta_\kappa(z) \geq 0$ and in a non-target region when $\Delta_\kappa(z) < 0$. Equivalently, its nearest anchor under cosine similarity belongs to the corresponding set. COMPASS uses this margin during generation to identify latents requiring update and during detection to recover the keyed signal from re-encoded text.

3.4 Selective Norm-Preserving Latent Steering

At each active denoising step, COMPASS identifies the latents that lie in non-target regions:

$$\mathcal{E}^{(s+1)} = \left\{ i \in \{1, \ldots, L\} : \Delta_\kappa\left(z_i^{(s+1)}\right) < 0 \right\}.$$

Latents whose indices are not in $\mathcal{E}^{(s+1)}$ already lie in a target region and require no correction. COMPASS selects $\lceil f\,|\mathcal{E}^{(s+1)}| \rceil$ of these latents, taking those with the smallest absolute margin, i.e. the ones closest to the decision boundary.

For each selected latent $z_i^{(s+1)}$, we first obtain its unit direction $u_i^{(s+1)} = u\left(z_i^{(s+1)}\right)$ using Eq. (1). We then sample a destination anchor uniformly from the target set:

$$a_i^\star \sim \mathrm{Uniform}\left(\mathcal{A}_\kappa^T\right).$$

Given the destination anchor $a_i^\star$, we steer $u_i^{(s+1)}$ toward it along the unit hypersphere using spherical linear interpolation. For a selected latent direction $u_i^{(s+1)}$ and sampled target anchor $a_i^\star$, let

$$\theta_i^{(s+1)} = \arccos\left(\left\langle u_i^{(s+1)}, a_i^\star \right\rangle\right)$$

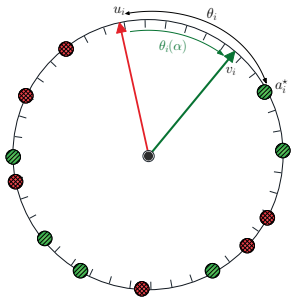


Figure 2: **Directional latent steering.** COMPASS steers u_i toward a target anchor $a_i^\star$, producing v_i.

denote their angular separation. The spherical linear interpolation used by COMPASS is:

$$v_i^{(s+1)} = \mathrm{SLERP}\Big(u_i^{(s+1)}, a_i^\star; \alpha\Big) = \frac{\sin\Big((1-\alpha)\theta_i^{(s+1)}\Big)}{\sin\Big(\theta_i^{(s+1)}\Big)} u_i^{(s+1)} + \frac{\sin\Big(\alpha\theta_i^{(s+1)}\Big)}{\sin\Big(\theta_i^{(s+1)}\Big)} a_i^\star.$$

Here, $\alpha \in [0, 1]$ controls the steering strength, which is the fraction of the geodesic traversed toward the target anchor. Increasing α moves the latent direction along the geodesic toward the target anchor, placing it directly on the anchor when $\alpha = 1$. The latent direction remains unchanged when $\alpha = 0$ or $\theta_i^{(s+1)} = 0$. As illustrated in Figure 2, the outer arc θ_i denotes the initial angular separation, while $\theta_i(\alpha)$ denotes the displacement induced by steering strength α. Green and red markers denote target and non-target anchors, respectively. The subsequent diversity perturbation is omitted for clarity.

After each steering operation, COMPASS adds a random perturbation $\varepsilon_i \sim \mathcal{N}(0, I_d)$ to prevent different latents from collapsing onto the same direction, especially for large values of α. To introduce local variation without directly opposing the watermark, we remove the component of ε_i that is parallel to the target anchor:

$$\eta_i^\perp = \frac{\varepsilon_i - \langle \varepsilon_i, a_i^\star \rangle \, a_i^\star}{\|\varepsilon_i - \langle \varepsilon_i, a_i^\star \rangle \, a_i^\star\|_2}.$$

The perturbed direction is then:

$$\widetilde{u}_i^{(s+1)} = \frac{v_i^{(s+1)} + \sigma\eta_i^\perp}{\left\| v_i^{(s+1)} + \sigma\eta_i^\perp \right\|_2},$$

where $\sigma \geq 0$ controls the strength of the noise added.

Finally, COMPASS restores the original magnitude of the latent:

$$\widetilde{z}_i^{(s+1)} = \left\| z_i^{(s+1)} \right\|_2 \widetilde{u}_i^{(s+1)}.$$

The move therefore changes the latent direction while preserving its norm.

At every active denoising step, COMPASS recomputes the keyed region assignment of each latent and samples a target anchor for every selected non-target latent. A previously corrected latent is therefore modified again only if a subsequent denoising step moves it back into a non-target region. Repeating this correction throughout the watermarking window reinforces the keyed signal while allowing the diffusion process to continue refining the representation.

Appx. D illustrates the steering operation and subsequent latent evolution.

3.5 Watermark Detection

Given a candidate text x, the detector passes the entire tokenized sequence through the model's frozen encoder, which returns a token-level latent representation:

$$Z_x^{\mathrm{enc}} = \left[z_{x,1}^{\mathrm{enc}}, \ldots, z_{x,L_x}^{\mathrm{enc}} \right]^\top \in \mathbb{R}^{L_x \times d},$$

where $z_{x,i}^{\mathrm{enc}}$ is the recovered latent for position i, and L_x is the number of non-padding latent positions in the candidate text. The length L_x need not equal the generation-unit size L or be divisible by it, since a candidate text may span several generation units.

For a supplied secret key κ, the detector reconstructs the corresponding target and non-target anchor sets and computes the keyed margin according to Eq. (2):

$$d_{\kappa,i} = \Delta_\kappa \left(z_{x,i}^{\mathrm{enc}} \right)$$

for each recovered latent. We summarize the candidate text by its mean keyed margin:

$$m_\kappa(x) = \frac{1}{L_x} \sum_{i=1}^{L_x} d_{\kappa,i}.$$

A larger value indicates a stronger preference for the target regions of the codebook derived from κ.

We standardize this statistic using the mean and standard deviation of $m_\kappa(x)$ over held-out unwatermarked text:

$$S_\kappa(x) = \frac{m_\kappa(x) - \mu^{\text{clean}}_{m,\kappa}}{\sigma^{\text{clean}}_{m,\kappa}}.$$

Here, $\mu^{\text{clean}}_{m,\kappa}$ and $\sigma^{\text{clean}}_{m,\kappa}$ denote the mean and standard deviation, respectively, of the clean mean-margin scores computed using the codebook associated with key κ.

We classify x as watermarked under key κ when $S_\kappa(x) \geq \lambda_\kappa$, where λ_κ is calibrated on held-out unwatermarked text at the desired false-positive rate. We report detection at 1% FPR and denote the corresponding true-positive rate by TPR@1% FPR.

4 Empirical Evaluation

4.1 Experimental Setup

Models and data. For Cola (Guo et al., 2026), we sample 600 documents from the `realnewslike` split of C4 (Raffel et al., 2020), following the sample size used for the evaluation of prior watermarking methods (Gloaguen et al., 2026). We retain only documents with at least 256 Cola tokens. We use a 128-token prefix as the prompt and generate a 128-token continuation. Cola uses 16-dimensional VAE latents, 16-token blocks, 16 denoising steps, and a guidance scale of 7 (Ho & Salimans, 2022). In contrast, ELF-L (Hu et al., 2026) generates text unconditionally. We therefore split each 1,024-token sample into eight 128-token segments and score each segment separately. For Cola, each watermarked continuation is paired with an unwatermarked continuation generated from the same prompt. For ELF-L, each watermarked segment is paired with the segment at the same position in an unwatermarked generation. The unwatermarked member of each pair serves as a calibration negative and a quality reference.

Baselines. We compare COMPASS against eight adapted watermarking methods. Six are token-level methods applied to the logits of the token decoder: KGW (Kirchenbauer et al., 2023), Unigram (Zhao et al., 2024), GumbelSoft (Fu et al., 2024), TextSeal (Sander et al., 2026), SynthID (Dathathri et al., 2024), and DiPmark (Wu et al., 2024). The other two, Tree-Ring (Wen et al., 2023) and Gaussian Shading (Yang et al., 2024), embed their keys in the initial latent noise and are detected by inverting the denoising process applied to the re-encoded text. We do not evaluate DDLM-specific watermarks because their embedding mechanisms rely on discrete states that are absent from continuous denoising. Gloaguen et al. (2026) modify categorical token distributions under partially resolved masked context, while dgMARK (Hong & No, 2026) steers the discrete order in which masked positions are revealed. Applying either mechanism to a CDLM would require redesigning the watermark rather than directly adapting the published method. Appx. F lists all hyperparameters, including the increased baseline strengths used when the default settings yield negligible detection.

Metrics. We calibrate the detection thresholds to a 1% FPR on unedited, unwatermarked texts and keep them fixed under every attack. *Deg.* is the relative percentage increase in the mean CompassJudger-2-7B (Zhang et al., 2025) degradation rating for watermarked text compared with its unwatermarked counterpart. The rating covers fluency, coherence, repetition, naturalness, and meaningfulness, each scored from 0 to 4. Similarly to prior work that measures watermark imperceptibility using the absolute perplexity gap between watermarked and clean generations (Kim et al., 2026), we report $|\Delta \log \mathrm{PPL}| = |\log \mathrm{PPL}_{\mathrm{wm}} - \log \mathrm{PPL}_{\mathrm{clean}}|$. This measures the magnitude of the deviation from the clean output rather than treating a decrease in perplexity as an improvement. Log-perplexity is computed using Llama-3.1-8B (Grattafiori et al., 2024) and is conditioned on the prompt for Cola (Appx. K).

Attacks. We apply eleven edits that require no access to the model, key, or detector (Appx. G). These comprise random deletion of 10% to 50% of the words (WD), deletion of a contiguous 30% span (Span30), retention of only the first or last half (First50 and Last50), adjacent-word swapping (Swap20), letter substitution (Typo5), sentence shuffling (Shuffle), and replacement of 25% or 50% of the sentences with their counterparts from the paired unwatermarked text (CP25 and CP50).

Table 1: **Detection, quality, and robustness on Cola.** TPR (%) at 1% FPR using a fixed threshold, evaluated on 600 watermarked texts both without attack and under the attacks described in Sec. 4.1. Deg. and $|\Delta \log \mathrm{PPL}|$ quantify the quality cost. The best value in each column is shown in **bold**, and the second best is underlined.

Method	Quality ↓		Unattacked ↑	Deletion and truncation ↑						Surface edits ↑			Dilution ↑	
	Deg. (%)	$\lvert\Delta \log \mathrm{PPL}\rvert$		WD10	WD30	WD50	Span30	First50	Last50	Swap20	Typo5	Shuffle	CP25	CP50
COMPASS (ours)	**9.1**	1.09	**94.2**	**86.8**	**88.3**	**82.3**	**86.2**	**77.2**	**86.0**	**89.7**	**78.7**	**88.2**	**77.3**	**54.3**
KGW	10.9	0.76	35.2	17.8	10.2	4.7	16.3	11.7	15.8	14.7	11.8	28.8	23.0	14.7
Unigram	11.6	0.89	35.2	24.8	20.5	13.2	17.8	10.3	17.5	27.8	22.3	29.8	22.7	13.5
GumbelSoft	22.4	1.50	49.5	21.8	4.7	1.3	25.2	16.8	21.2	14.3	9.8	39.7	32.8	23.8
TextSeal	9.2	**0.72**	76.8	47.8	22.2	6.2	52.5	37.5	46.5	44.7	29.3	70.0	65.3	50.2
SynthID	15.0	1.07	72.0	26.5	7.0	1.8	35.3	25.3	29.5	17.2	12.0	55.5	51.3	30.8
DiPmark	15.9	1.06	8.3	6.7	4.7	2.3	6.5	4.3	3.5	4.2	3.5	7.7	5.3	3.5
Tree-Ring	25.9	2.41	89.2	2.0	0.0	0.0	5.5	0.0	0.0	79.0	5.2	14.7	38.0	20.2
Gaussian Shading	34.2	1.29	31.5	32.5	45.0	45.8	43.8	42.7	50.0	31.5	37.3	33.7	26.8	23.0

Table 2: **Detection, quality, and robustness on ELF-L.** We use the same evaluation protocol and notation as in Table 1. Each evaluated text is a 128-token segment of an unconditional 1,024-token generation.

Method	Quality ↓		Unattacked ↑	Deletion and truncation ↑						Surface edits ↑			Dilution ↑	
	Deg. (%)	$\lvert\Delta \log \mathrm{PPL}\rvert$		WD10	WD30	WD50	Span30	First50	Last50	Swap20	Typo5	Shuffle	CP25	CP50
COMPASS (ours)	4.3	0.62	**100.0**	**100.0**	**100.0**	**99.7**	**100.0**	**99.8**	**100.0**	**100.0**	**93.0**	**100.0**	**99.5**	**70.0**
KGW	8.3	0.82	75.5	65.8	44.5	20.5	59.2	46.0	45.3	57.8	23.3	74.0	49.3	23.3
Unigram	6.5	1.11	83.2	80.8	73.3	60.8	73.7	61.2	61.5	83.2	70.7	83.2	61.5	31.5
GumbelSoft	8.8	0.72	83.7	68.7	36.7	15.0	73.0	61.8	67.5	60.3	34.5	80.2	71.7	48.2
TextSeal	7.5	0.67	77.8	61.5	33.5	10.5	67.3	56.0	59.8	52.0	29.5	74.7	64.2	41.5
SynthID	9.7	0.91	36.7	28.8	19.5	7.3	17.7	7.8	7.8	36.7	57.5	36.7	17.0	6.2
DiPmark	5.2	2.51	92.0	59.3	13.3	2.3	76.3	63.7	58.3	35.0	7.0	87.3	79.3	69.7
Tree-Ring	31.9	**0.17**	18.2	9.2	26.0	11.5	26.2	16.8	12.8	15.0	6.5	12.0	12.3	11.2
Gaussian Shading	**2.4**	1.92	53.5	53.5	58.2	63.7	58.2	64.7	63.3	53.8	75.5	53.5	30.7	14.7

4.2 RESULTS

COMPASS offers the best trade-off between detection and quality. On Cola, COMPASS detects 94.2% of watermarked texts with the lowest judged degradation of any method, at 9.1% (Table 1). Tree-Ring comes closest in detection performance, reaching 89.2%, but incurs a substantially higher quality cost of 25.9%. No token-level baseline exceeds 76.8% detection. On ELF-L, COMPASS detects every watermarked text with only 4.3% degradation (Table 2). The strongest token-level baselines are DiPmark, GumbelSoft, and Unigram, with TPRs of 92.0%, 83.7%, and 83.2%, respectively, and all incur greater judged degradation than COMPASS. Gaussian Shading has a lower cost of 2.4% but detects only 53.5% of the watermarked texts. The Cola per-method frontiers in Figure 3 place COMPASS in the favorable high-detection, low-degradation region.

COMPASS survives edits that erase the baselines' signal. Deleting only 10% of the words (WD10) in Table 1 reduces Tree-Ring's TPR on Cola from 89.2% to 2.0%, while retaining only the last half reduces it to 0.0%. Under the same edits, COMPASS retains TPRs of 86.8% and 86.0%, respectively. Tree-Ring's Fourier-domain key appears to require positional alignment, which deletion disrupts but local swapping largely preserves, as reflected by its 79.0% TPR under Swap20. In contrast, COMPASS averages margins across latents and does not require positional alignment. It leads under all eleven attacks on Cola, retaining 82.3% TPR after 50% word deletion, compared with at most 45.8% for any baseline. Clean-text replacement is the strongest attack on both models: CP50 reduces the TPR of COMPASS to 54.3% on Cola and 70.0% on ELF-L by diluting the aggregate margin with unwatermarked content. Apart from CP50, COMPASS retains at least 99.5% TPR under every attack on ELF-L except Typo5, for which it reaches 93.0% (Table 2).

Paraphrasing and ablations. Under DIPPER 40/40 paraphrasing (Krishna et al., 2023), COMPASS achieves TPRs of 94.5% on ELF-L and 46.3% on Cola, compared with 8.3% and 20.0% for the strongest token-level baselines, respectively (Table 3). The strengthened baseline configurations

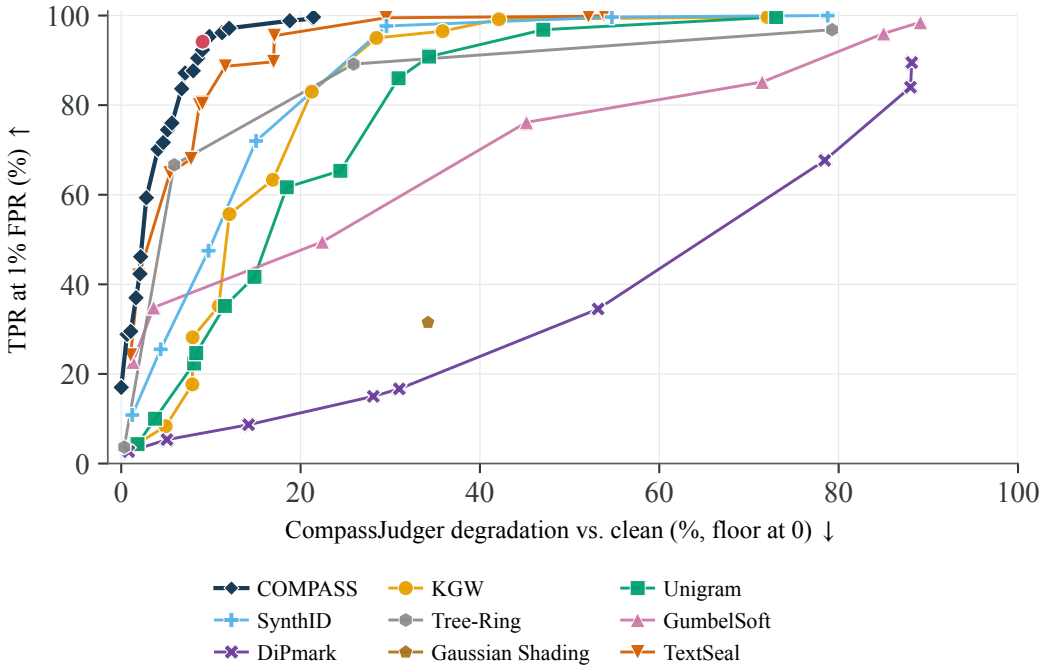


Figure 3: **Detection–quality trade-offs on Cola.** Lines connect the best evaluated settings for each method. The red dot marks the COMPASS setting used in our main evaluation.

used in this comparison are listed in Appx. F.5. Appx. H reports ablations of the COMPASS hyperparameters, and Appx. J examines whether differences in the models' representation spaces may help explain their different levels of paraphrase robustness.

Table 3: **Robustness to DIPPER paraphrasing.** TPR (%) at 1% FPR on Cola (left) and ELF-L (right). Column headings denote the lexical/order diversity settings. "None" denotes unattacked text. The best result in each column is shown in **bold**, and the second best is underlined.

(a) Cola

Method	None	20/20	40/40	60/60	80/80
COMPASS	94.2	**62.3**	**46.3**	**34.3**	**20.0**
KGW	93.3	12.2	10.7	6.2	3.0
Unigram	**96.5**	28.3	20.0	12.0	6.3
GumbelSoft	94.0	3.8	2.7	1.7	1.2
TextSeal	93.5	7.5	4.5	3.3	1.2

(b) ELF-L

Method	None	20/20	40/40	60/60	80/80
COMPASS	**100.0**	**99.2**	**94.5**	**77.8**	**44.0**
KGW	97.5	1.8	1.7	0.8	1.0
Unigram	99.7	13.8	8.3	5.7	3.3
GumbelSoft	96.0	3.0	2.8	1.7	2.0
TextSeal	**100.0**	3.5	1.2	0.8	0.0

5 Conclusions

We introduced COMPASS, the first watermark purpose-built for continuous diffusion language models. Existing watermarking schemes operate on token probabilities or discrete masking states, neither of which provides a natural interface for the continuous generation process of CDLMs. COMPASS instead embeds a key-dependent directional signal directly into the denoising trajectory. At detection time, the candidate text is re-encoded and scored for the same keyed preference. The method is training-free, leaves the model parameters unchanged, and adds no neural-network evaluations during generation. Empirically, COMPASS achieves the highest detection rate on both models and leads under every post-generation edit we evaluate. It also incurs the lowest judged quality degradation on Cola and the second lowest on ELF-L. Together, these results make robust provenance tracing practical for CDLMs.

REFERENCES

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=h7-XixPCAL.

Ruibo Chen, Yihan Wu, Yanshuo Chen, Chenxi Liu, Junfeng Guo, and Heng Huang. A watermark for order-agnostic language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Nlm3Xf0W9S.

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, Jamie Hayes, Nidhi Vyas, Majd Al Merey, Jonah Brown-Cohen, Rudy Bunel, Borja Balle, Taylan Cemgil, Zahra Ahmed, Kitty Stacpoole, Ilia Shumailov, Ciprian Baetu, Sven Gowal, Demis Hassabis, and Pushmeet Kohli. Scalable watermarking for identifying large language model outputs. *Nature*, 634(8035):818–823, 2024. doi: 10.1038/s41586-024-08025-4. URL https://www.nature.com/articles/s41586-024-08025-4.

Zhicheng Du and Lan Ma. Continuous language diffusion as a decoder-interface problem, 2026. URL https://arxiv.org/abs/2606.08810.

Jiayi Fu, Xuandong Zhao, Ruihan Yang, Yuansen Zhang, Jiangjie Chen, and Yanghua Xiao. GumbelSoft: Diversified language model watermarking via the GumbelMax-trick. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5791–5808, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.315. URL https://aclanthology.org/2024.acl-long.315/.

Enoal Gesny, Eva Giboulot, Teddy Furon, and Vivien Chappelier. Guidance watermarking for diffusion models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=5ifzhjMCKq.

Angeliki Giannou, Shashank Rajput, Jy-Yong Sohn, Kangwook Lee, Jason D. Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11398–11442. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/giannou23a.html.

Thibaud Gloaguen, Robin Staab, Nikola Jovanović, and Martin Vechev. Watermarking diffusion language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=3aBWTYGcaT.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley

Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu,

Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Hongcan Guo, Qinyu Zhao, Yian Zhao, Shen Nie, Rui Zhu, Qiushan Guo, Feng Wang, Tao Yang, Hengshuang Zhao, Guoqiang Wei, and Yan Zeng. Continuous latent diffusion language model, 2026. URL https://arxiv.org/abs/2605.06548.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Pyo Min Hong and Albert No. dgMARK: Decoding-guided watermarking for diffusion language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2601.22985.

Abe Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. SemStamp: A semantic watermark with paraphrastic robustness for text generation. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4067–4082, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.226. URL https://aclanthology.org/2024.naacl-long.226/.

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. URL https://arxiv.org/abs/2605.10938.

Jaehyeong Jo and Sung Ju Hwang. Continuous diffusion model for language modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=VGv5y60sXC.

Zakhar Kabluchko and Christoph Thaele. The typical cell of a voronoi tessellation on the sphere, 2021. URL https://arxiv.org/abs/1911.07221.

Jungin Kim, Shinwoo Park, and Yo-Sub Han. Marking code without breaking it: Code watermarking for detecting llm-generated code, 2026. URL https://arxiv.org/abs/2502.18851.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A Watermark for Large Language Models. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 27469–27500. Curran Associates, Inc., 2023. doi: 10.52202/075280-1195. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/575c450013d0e99e4b0ecf82bd1afaa4-Paper-Conference.pdf.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=FpaCL1MO2C.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori Hashimoto. Diffusion-LM improves controllable text generation. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=3s9IrEsjLyk.

Jiacheng Liang, Zian Wang, Spencer Hong, Shouling Ji, and Ting Wang. Watermark under fire: A robustness evaluation of LLM watermarking. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 21050–21074, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1148. URL https://aclanthology.org/2025.findings-emnlp.1148/.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=CNicRIVIPA.

Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Seo Shekhtman, and Kilian Q Weinberger. Latent diffusion for language generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

K.V. Mardia and P.E. Jupp. *Directional Statistics*. Wiley Series in Probability and Statistics. Wiley, 2000. ISBN 9780471953333. URL https://books.google.de/books?id=zjPvAAAAMAAJ.

George A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995. ISSN 0001-0782. doi: 10.1145/219717.219748. URL https://doi.org/10.1145/219717.219748.

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. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M Rush, Yair Schiff, Justin T Chiu, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Tom Sander, Hongyan Chang, Tomáš Souček, Tuan Tran, Valeriu Lacatusu, Sylvestre-Alvise Rebuffi, Alexandre Mourachko, Surya Parimi, Christophe Ropers, Rashel Moritz, Vanessa Stark, Hady Elsahar, and Pierre Fernandez. Textseal: A localized llm watermark for provenance and distillation protection. *arXiv preprint arXiv:2605.12456*, 2026.

Ken Shoemake. Animating rotation with quaternion curves. *SIGGRAPH Comput. Graph.*, 19(3): 245–254, July 1985. ISSN 0097-8930. doi: 10.1145/325165.325242. URL https://doi.org/10.1145/325165.325242.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von

Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Z57JrmubNl.

Yihan Wu, Zhengmian Hu, Junfeng Guo, Hongyang Zhang, and Heng Huang. A resilient and accessible distribution-preserving watermark for large language models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=c8qWiNiqRY.

Zijin Yang, Kai Zeng, Kejiang Chen, Han Fang, Weiming Zhang, and Nenghai Yu. Gaussian shading: Provable performance-lossless image watermarking for diffusion models. In *CVPR*, pp. 12162–12171, 2024. URL https://doi.org/10.1109/CVPR52733.2024.01156.

Taolin Zhang, Maosong Cao, Alexander Lam, Songyang Zhang, and Kai Chen. Compassjudger-2: Towards generalist judge model via verifiable rewards, 2025. URL https://arxiv.org/abs/2507.09104.

Xuandong Zhao, Prabhanjan Vijendra Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for AI-generated text. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=SsmT8aO45L.

Cai Zhou, Chenxiao Yang, Yi Hu, Chenyu Wang, Chubin Zhang, Muhan Zhang, Lester Mackey, Tommi Jaakkola, Stephen Bates, and Dinghuai Zhang. Coevolutionary continuous discrete diffusion: Make your diffusion language model a latent reasoner. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=80nuPd5Gvb.

A Token-Output Distribution Analysis

Our experiments show that watermarks operating on token probabilities perform poorly on CDLMs. At almost every generated position, CDLMs effectively determine the token to emit. This leaves no room for a logit-level watermark to act. We measure this property for both Cola and ELF-L.

Measurement protocol. The two models cannot be run under identical conditions. Cola generates continuations, so we prompt it with 128 tokens from the RealNewsLike subset of C4 and generate 256 tokens for each of 256 documents, yielding 65,536 positions. ELF-L is an unconditional sampler and accepts no prompt, so we collect 256 documents of 1024 tokens and discard all positions from the first EOS token onward, leaving 261,849 positions. For both models we record the logits the decoder commits to, before any sampling transform is applied, and we process them with the same reduction code, which is what makes the two measurements directly comparable. The reduction is performed in float64. In float32, the log-sum-exp underflows once the competing probability mass falls below machine epsilon, and ELF-L would appear exactly one-hot at most of its positions.

Let $\ell_i \in \mathbb{R}^{|\mathcal{V}|}$ denote the output logits at generated position i, where $\mathcal{V}$ is the model vocabulary. We convert these logits into a token distribution

$$p_i = \mathrm{softmax}(\ell_i)$$

and measure its maximum probability

$$p_{\max,i} = \max_{v \in \mathcal{V}} p_i(v)$$

and entropy, in bits,

$$H_i = -\sum_{v \in \mathcal{V}} p_i(v) \log_2 p_i(v).$$

The maximum probability measures how strongly the model favors its most likely token, while the entropy measures how broadly probability mass is distributed across the vocabulary. A distribution with $p_{\max,i}$ close to one and H_i close to zero is effectively concentrated on a single token.

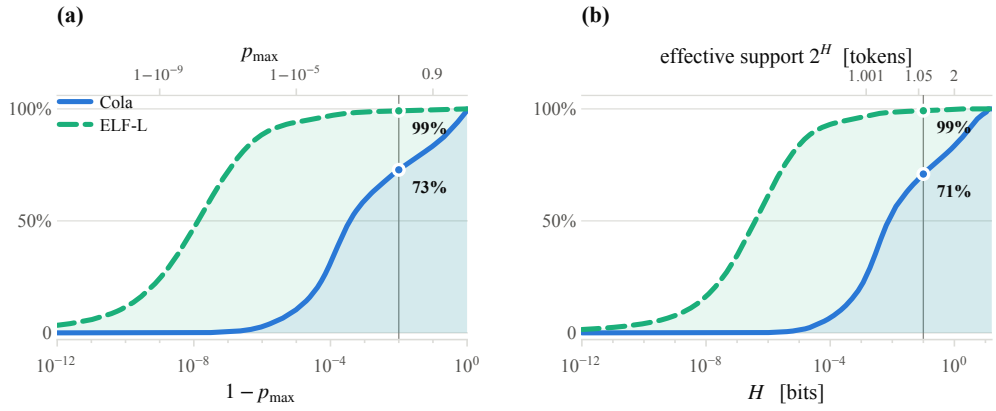


Figure 4: **Token distributions of CDLMs are close to one-hot. (a)** Empirical CDF of $1 - p_{\max}$, with $p_{\max}$ shown on the upper axis. The vertical rule marks $p_{\max} = 0.99$. **(b)** The same measurement in Shannon entropy, with the upper axis giving the effective support 2^H, the number of equally likely tokens a position is worth. The vertical rule marks $H = 0.1$ bit. Positions whose value falls below the plotted range are pinned at the left edge rather than discarded, so each curve remains exact at every plotted value of x. This affects 3.4% of ELF-L positions in (a) and 1.5% in (b).

Both models are near-deterministic at the token level. Figure 4(a) reports the empirical distribution of $1 - p_{\max}$, the amount of probability mass left for every token other than the most likely one. Its median is 1.3×10^{-8} for ELF-L and 3.7×10^{-4} for Cola, so the median $p_{\max}$ is $1 - 1.3 \times 10^{-8}$ and 0.9996 respectively. These two values differ by four orders of magnitude, yet both indicate that the runner-up token is essentially never competitive. Figure 4(b) restates the measurement in terms of entropy, and its upper axis converts entropy into effective support, which counts how many

equally likely tokens a distribution is worth. At the median position this count is 1.0000 for ELF-L and 1.0052 for Cola, so each model is in practice selecting among a single candidate. We refer to distributions of this kind as *near-one-hot*. The term describes the normalized token probabilities and not the raw logits, which remain unbounded real-valued scores. We note that H and $1 - p_{\max}$ are almost perfectly correlated in our measurements (Pearson correlation of 0.998 for Cola and 0.9997 for ELF-L on the logged quantities).

A statistic pooled over all positions can conceal systematic variation. We therefore also measure how token-level confidence depends on the position within a sequence and on the length of the requested output.

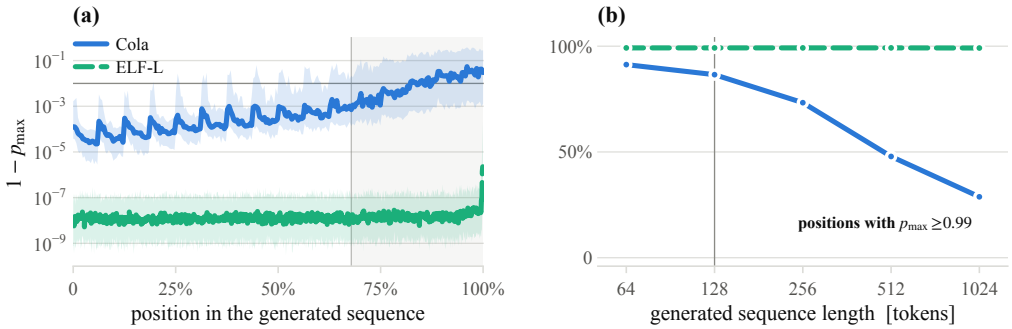


Figure 5: **Cola becomes less confident as its output grows; ELF-L does not.** **(a)** Per-position median of $1 - p_{\max}$ with an interquartile band, plotted against relative position so that both models span the axis. Medians supported by fewer than 200 documents are omitted, which removes one ELF-L tail position. The shaded region marks the part of Cola's output that has degenerated into repetition: beyond token 173 of 256, more than half of its tokens repeat one of the previous eight. **(b)** Share of positions with $p_{\max} \geq 0.99$ as a function of the length of the generated text. The vertical rule marks the 128-token setting used in our main experiments. Both curves are read the same way, as prefixes of a single 1024-token run per model, because the ELF-L sampler has no length parameter and cannot be regenerated at each length.

Confidence decays along Cola's sequence. Figure 5(a) plots the per-position median of $1 - p_{\max}$ against relative position. The ELF-L curve is flat, changing by a factor of 1.7 between the first and last tenth of the sequence. The Cola curve rises by a factor of 498 over the same span and crosses the $p_{\max} = 0.99$ rule partway through. This decay is a property of the sampled text rather than of the decoder. Over the same span, the share of Cola positions whose token repeats one of the previous eight rises from 10.0% to 74.3%, and across positions this share correlates 0.936 with the plotted median. The corresponding ELF-L share is flat, moving only from 7.0% to 7.3% and correlating 0.27. The decay is thus confined to the shaded region, where Cola has fallen into repetition; throughout the part of the sequence in which the model still produces usable text, its decoder remains near-deterministic. The regular oscillation visible on the Cola curve has a period of exactly 16 tokens, which matches its VAE block size. Part of the variation in the panel is therefore attributable to position within a block rather than to the trend across the sequence; the decile statistics reported above are medians over the first and last tenth of positions, which span more than one full block and so average the oscillation out rather than sampling it at a single phase, and we measure the within-block profile separately with each block's own level divided out.

Confidence also decays with the requested output length. Figure 5(b) reports the share of positions with $p_{\max} \geq 0.99$ as the output length varies. The ELF-L curve is flat across the full range, from 99.2% at 64 tokens to 99.1% at 1024. The Cola curve declines steadily, from 91.3% to 28.8% over the same range; its median $p_{\max}$ falls from 0.99993 at 64 tokens to 0.65 at 1024 tokens.

Because both curves are read from prefixes of one long run rather than from a separate generation per length, we verified that this construction does not drive the result. Cola supports both methods, and at every length on the ladder the prefix of its 1024-token run agrees with a dedicated run generated at that target length to within one percentage point of the plotted share; at 256 tokens the two medians agree to three decimal places. The confidence at a given position therefore does not depend, to within that margin, on how long the requested output is; a longer request lowers the pooled statistic

only by adding later, less confident positions. The near-one-hot regime is thus delimited by an absolute position rather than by a fraction of the output. Our experiments generate 128 tokens per continuation, marked by the vertical rule, and consequently operate well within it. This is also why we fix the evaluation length at 128 tokens for every method compared in this paper. Generating longer outputs would make the comparison unsound in two ways at once. Text quality would be scored on generations that have degenerated regardless of any watermark, so the quality axis would stop measuring the watermark's cost; and the additional positions would be the low-confidence ones, which would inflate the entropy available to logit-level watermarks and overstate their detection performance. Restricting evaluation to 128 tokens removes both effects and is the more demanding setting for our own method: at that length 86.5% of Cola positions satisfy $p_{\max} \geq 0.99$, against 72.8% when measured over 256 tokens, so the measurement in Figure 4 understates how concentrated the model is in the regime we actually evaluate.

Implication for watermark design. A watermark that biases the output distribution requires usable entropy at the point of token selection, and our experiments show that CDLMs have a low entropy per token. When a single token holds nearly all the probability mass, a small perturbation of the logits rarely changes which token is sampled, and the watermark leaves no detectable trace. A perturbation large enough to make an alternative token competitive alters the output the model would otherwise have produced and substantially degrades text quality. The positions at which Cola retains usable entropy do not resolve this, because they lie almost entirely outside the operating regime: in a 1024-token generation, 97.6% of the positions with $p_{\max} < 0.99$ occur beyond token 128, where the rate of such positions is 79.5% against 13.5% within the first 128, and where Figure 5(a) shows the output has degenerated into repetition. Within the 128 tokens our experiments generate, only 13.5% of positions fall below $p_{\max} = 0.99$ and 7.5% below 0.9, which is too little entropy, and too unevenly distributed, to carry a detectable signal.

B Watermark Injection and Detection Details

This appendix provides the algorithms for watermark injection and detection.

B.1 Watermark Injection

Algorithm 1 summarizes the complete watermark-injection procedure described in Sec. 3.4.

B.2 Watermark Detection

Algorithm 2 summarizes detection for a supplied secret key. The clean reference statistics and detection threshold are calibrated beforehand, as described in Sec. 3.5.

Algorithm 1: COMPASS Watermark Injection

Input: Initial latent state $Z^{(0)}$, denoising transitions $\{\Phi_s\}_{s=0}^{S-1}$, keyed anchor sets $\mathcal{A}_\kappa^T$ and $\mathcal{A}_\kappa^N$, watermarking window $[s_{\text{start}}, s_{\text{end}}]$, selection fraction f, steering strength α, and noise strength σ

Output: Final latent state $Z^{(S)}$

for $s \leftarrow 0$ **to** $S-1$ **do**
  if $s_{\text{start}} \le s \le s_{\text{end}}$ **then**
    $\mathcal{E}^{(s)} \leftarrow \left\{ i \in \{1,\dots,L\} : \Delta_\kappa\left(z_i^{(s)}\right) < 0 \right\}$
    Retain in $\mathcal{E}^{(s)}$ the $\lceil f\,|\mathcal{E}^{(s)}| \rceil$ indices with the smallest values of $\left|\Delta_\kappa\left(z_i^{(s)}\right)\right|$
    if $\mathcal{E}^{(s)} \neq \varnothing$ **then**
      foreach $i \in \mathcal{E}^{(s)}$ **do**
        $u_i^{(s)} \leftarrow \text{normalize}\left(z_i^{(s)}\right)$
        $a_i^\star \sim \text{Uniform}\left(\mathcal{A}_\kappa^T\right)$
        $v_i^{(s)} \leftarrow \text{SLERP}\left(u_i^{(s)}, a_i^\star; \alpha\right)$
        $\varepsilon_i \sim \mathcal{N}(0, I_d)$
        $\eta_i^\perp \leftarrow \text{normalize}(\varepsilon_i - \langle \varepsilon_i, a_i^\star \rangle\, a_i^\star)$
        $\widetilde{u}_i^{(s)} \leftarrow \text{normalize}\left(v_i^{(s)} + \sigma \eta_i^\perp\right)$
        $\widetilde{z}_i^{(s)} \leftarrow \left\| z_i^{(s)} \right\|_2 \widetilde{u}_i^{(s)}$
      $\widetilde{Z}_{j,:}^{(s)} \leftarrow \begin{cases} \widetilde{z}_j^{(s)}, & j \in \mathcal{E}^{(s)}, \\ z_j^{(s)}, & j \notin \mathcal{E}^{(s)}, \end{cases} \quad j \in \{1,\dots,L\}$
      $Z^{(s+1)} \leftarrow \Phi_s\left(\widetilde{Z}^{(s)}\right)$
    else
      $Z^{(s+1)} \leftarrow \Phi_s\left(Z^{(s)}\right)$
  else
    $Z^{(s+1)} \leftarrow \Phi_s\left(Z^{(s)}\right)$
return $Z^{(S)}$

Algorithm 2: COMPASS watermark detection

Input: Candidate text x, secret key κ, model-specific reference pool $\mathcal{B}_M$, frozen encoder E, clean reference statistics $\mu_{m,\kappa}^{\text{clean}}$ and $\sigma_{m,\kappa}^{\text{clean}}$, and detection threshold λ_κ

Output: Detection score $S_\kappa(x)$ and decision $\widehat{y} \in \{0,1\}$

$(\mathcal{A}_\kappa^T, \mathcal{A}_\kappa^N) \leftarrow \text{ConstructAnchors}(G_\kappa, \mathcal{B}_M)$
$Z_x^{\text{enc}} \leftarrow E(x)$
Remove padding positions from Z_x^{enc} and let L_x denote the number of remaining positions
for $i \leftarrow 1$ **to** L_x **do**
  $d_{\kappa,i} \leftarrow \Delta_\kappa\left(z_{x,i}^{\text{enc}}\right)$
$m_\kappa(x) \leftarrow \dfrac{1}{L_x} \displaystyle\sum_{i=1}^{L_x} d_{\kappa,i}$
$S_\kappa(x) \leftarrow \dfrac{m_\kappa(x) - \mu_{m,\kappa}^{\text{clean}}}{\sigma_{m,\kappa}^{\text{clean}}}$
$\widehat{y} \leftarrow \mathbb{1}[S_\kappa(x) \ge \lambda_\kappa]$
return $S_\kappa(x), \widehat{y}$

C Latent Distributions and Anchor Sources

Figure 6 tracks the distributions of individual latent coordinates across generation steps. Each curve is a normalized density obtained by pooling the scalar entries of latent vectors at one step. Both models remain approximately centered at zero, but their trajectories differ. In Cola, the first-step distribution has a pronounced central peak. As denoising proceeds, that peak falls and more density appears away from zero. The dashed VAE encoder curve is approximately symmetric and bell-shaped; at zero, its density lies below the first-step curve and above the final-step curve. ELF-L follows the opposite progression: its initially broad distribution becomes increasingly concentrated near zero, with the sharpest peak occurring after the final sampling update. Thus, the coordinate distribution depends on both the architecture and the stage of generation.

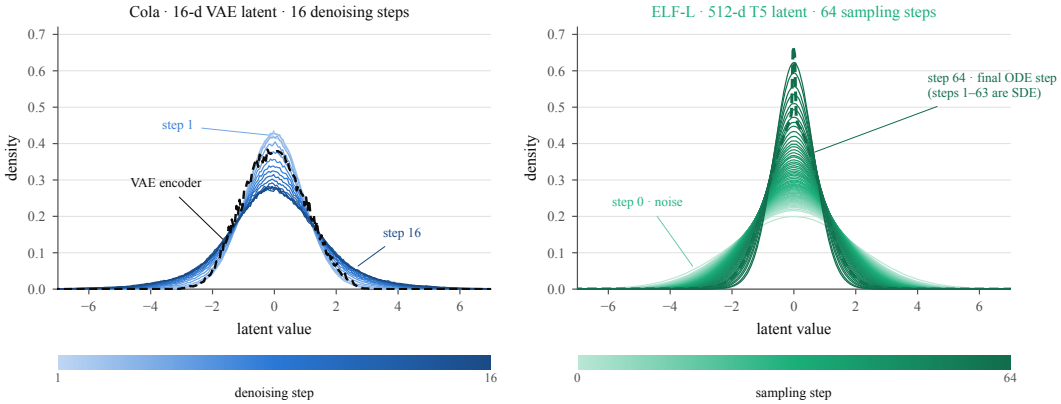


Figure 6: **Evolution of intermediate latent distributions.** Normalized densities of pooled latent coordinates at successive generation steps; lighter colors indicate earlier steps. The dashed curve in the Cola panel shows VAE encoder representations. Cola's distribution broadens during denoising, whereas ELF-L's becomes more concentrated near zero.

For either model, we could construct anchors by generating reference text with the model, encoding that text, and normalizing the resulting vectors. We use this procedure for ELF-L. For Cola, we instead draw and normalize Gaussian vectors, avoiding the need to generate and encode a reference corpus. The watermark injection and detection procedures are otherwise unchanged.

Cola. Cola's Text VAE uses a Gaussian encoder posterior and is trained with KL regularization toward a base prior (Guo et al., 2026). Its VAE encoder coordinates also have the approximately centered, symmetric profile shown in Figure 6. These properties motivate a simple isotropic anchor source in Cola's 16-dimensional latent space. Using the pseudorandom generator G_κ initialized by the secret key, we draw K candidate vectors and normalize each one:

$$\mathcal{D}_{\mathrm{Cola}} = \mathcal{N}(0, I_{16}), \qquad b_k \sim \mathcal{D}_{\mathrm{Cola}} \;\; (k = 1, \ldots, K), \qquad a_k = \frac{b_k}{\|b_k\|_2}.$$

The vectors b_k are Gaussian draws, but their normalized directions a_k are uniform on the unit sphere: the isotropic Gaussian favors no direction. Once the key is fixed, G_κ reproduces the same draws, so generation and detection reconstruct the same codebook. This requires no reference-text collection, VAE encoding pass, or shared pool of representations. Gaussian sampling simplifies and accelerates anchor construction for Cola.

ELF-L. ELF-L operates in a 512-dimensional T5 (Raffel et al., 2020) representation space. For this model, we use representations produced by its encoder. Let

$$\mathcal{B}_{\mathrm{ELF-L}} = \{c_1, \ldots, c_N\} \subset \mathbb{R}^{512}$$

be a fixed, ordered pool of token-level T5 encoder representations. We construct the pool by generating reference text with ELF-L itself, tokenizing that text, and passing the token sequences through its T5 encoder. The reference texts are separate from those used to evaluate the watermark. Every pool entry is eligible for selection with equal probability, so $\mathcal{D}_{\mathrm{ELF-L}} = \mathrm{Uniform}(\mathcal{B}_{\mathrm{ELF-L}})$.

For each key, G_κ selects K distinct pool entries without replacement and normalizes their vectors:

$$(i_1, \ldots, i_K) = \text{SampleWithoutReplacement}(G_\kappa, \{1, \ldots, N\}, K), \qquad b_k = c_{i_k}, \qquad a_k = \frac{b_k}{\|b_k\|_2}.$$

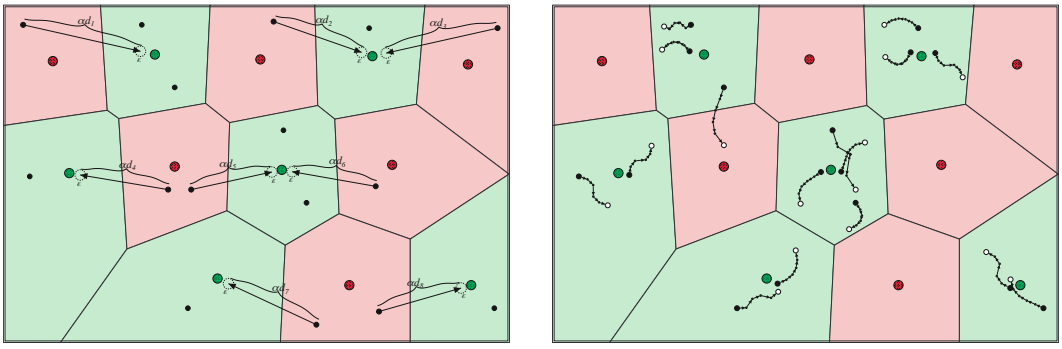
Prior analysis reports alignment between generated ELF states and T5 token-embedding geometry, together with direction-dependent behavior at the decoder interface (Du & Ma, 2026). This motivates drawing anchors from representations attained by the encoder rather than adopting an isotropic source for ELF-L by default. Because the pool and its ordering are fixed, generation and detection select and partition the same anchors from the same key.

These choices instantiate the model-specific reference distribution $\mathcal{D}_M$ in Sec. 3.3. The keyed partition into target and non-target anchors, together with the subsequent injection and detection procedures, is unchanged.

D Conceptual Illustration of Latent Steering

To provide intuition for the operation of COMPASS, we use hand-drawn two-dimensional illustrations of the latent-space geometry (Figure 7). The illustrations are informed by the qualitative behavior observed in our experiments, but they are not direct visualizations of measured latent representations, recorded denoising trajectories, or projections of the model's high-dimensional latent space.

For readability, the illustrations make one deliberate simplification: the illustrated destinations are the nearest target anchors, which avoids the long, intersecting trajectories that would obscure the diagram, whereas the actual method samples each destination uniformly from the target-anchor set. The illustrated movements therefore represent possible realizations of the steering operation rather than the destination-selection rule of COMPASS. Panel (a) shows every latent in a non-target region being corrected, which corresponds to the selection fraction $f = 1$; the method is defined for any $f \in (0, 1]$, and smaller values leave a subset of eligible latents unchanged at each active step.



(a) Conceptual illustration of latent steering. (b) Conceptual illustration of subsequent denoising.

Figure 7: **Conceptual view of latent steering and subsequent denoising.** The colored cells provide a simplified two-dimensional representation of the regions induced by target (green points) and non-target (red points) anchors. **(a)** Non-target latents are steered toward the depicted target anchors and receive a small diversity perturbation. The panel depicts the nearest target anchors as destinations; the actual method samples destination anchors uniformly from the target set. Every eligible latent is corrected, corresponding to $f = 1$. **(b)** The remaining denoising steps may continue moving the corrected latents through the partitioned space. The positions and trajectories are hand-drawn conceptual representations and do not correspond to directly measured model latents.

These illustrations highlight that COMPASS introduces a directional preference rather than permanently attaching a latent to an anchor. After the intervention, the model continues its ordinary denoising process and may move the corrected representations away from their sampled target anchors. The watermark only requires a sufficient preference for the keyed target regions to remain detectable after the remaining denoising steps and after the generated text has been decoded and re-encoded.

E WHY IS THE COMPASS WATERMARK DETECTABLE?

E.1 TERMINAL SURVIVAL OF THE ANCHOR WATERMARK

Let $d \geq 2$ and let

$$a_1, a_2, \ldots, a_K \overset{\text{iid}}{\sim} \text{Unif}(\mathbb{S}^{d-1})$$

be the unit-norm directions of a secret anchor codebook. Since all anchors have an equal norm, the nearest-anchor assignment depends only on the normalized latent direction. For any $u \in \mathbb{S}^{d-1}$, define

$$J(u) = \arg\max_{1 \leq k \leq K} a_k^\top u.$$

Ties have probability zero under the continuous anchor distribution.

At a chosen denoising step and position, COMPASS moves a latent direction toward a desired anchor while preserving its original latent norm. To isolate the geometric mechanism governing the survival of the watermark, we consider an exact-anchor intervention in which a source direction a_1 is inserted in the watermark step. Let $Y \in \mathbb{S}^{d-1}$ be the normalized final latent direction after all remaining denoising updates. Conditioned on the injected source anchor and generation randomness, we assume that Y is independent of the remaining $K-1$ competing anchors. This is the setting of our random-anchor survival experiment, in which the denoising model receives the injected source direction but does not subsequently query the competing anchors.

Define the terminal angular displacement and its equivalent chord distance by

$$\Theta = \arccos(a_1^\top Y), \qquad \Delta_{\text{ch}} = \|Y - a_1\|_2, \qquad \cos\Theta = 1 - \frac{\Delta_{\text{ch}}^2}{2}.$$

Let $C_d(\theta)$ denote the normalized surface area of a spherical cap with angular radius θ on $\mathbb{S}^{d-1}$:

$$C_d(\theta) = \begin{cases} \frac{1}{2} I_{\sin^2\theta}\left(\frac{d-1}{2}, \frac{1}{2}\right), & 0 \leq \theta \leq \frac{\pi}{2}, \\ 1 - \frac{1}{2} I_{\sin^2\theta}\left(\frac{d-1}{2}, \frac{1}{2}\right), & \frac{\pi}{2} < \theta \leq \pi, \end{cases}$$

where $I_x(a, b)$ is the regularized incomplete beta function (Mardia & Jupp, 2000). The spherical Voronoi interpretation of the anchor cells follows the standard geometry of Voronoi tessellations on the sphere (Kabluchko & Thaele, 2021).

Theorem 1 (Exact terminal source-cell survival) *Conditional on $\Theta = \theta$, the probability that the final latent remains in the Voronoi cell of its injected source anchor is*

$$\Pr[J(Y) = 1 \mid \Theta = \theta] = [1 - C_d(\theta)]^{K-1}.$$

Equivalently, the probability of terminal source-cell exit is

$$\Pr[J(Y) \neq 1 \mid \Theta = \theta] = 1 - [1 - C_d(\theta)]^{K-1}.$$

Proof: A competing anchor a_k defeats the source anchor at the final direction Y if and only if

$$a_k^\top Y \geq a_1^\top Y = \cos\theta.$$

Equivalently, a_k lies in the spherical cap centered at Y with angular radius θ. Each independent competing anchor lies outside this cap with probability $1 - C_d(\theta)$. Therefore, all $K-1$ competing anchors lie outside the cap with probability $[1 - C_d(\theta)]^{K-1}$. □

The result can be evaluated directly from the measured terminal chord displacement. Define

$$F_d(c) = I_{(c+1)/2}\left(\frac{d-1}{2}, \frac{d-1}{2}\right).$$

Since $F_d(c) = 1 - C_d(\arccos c)$, terminal source-cell exit can equivalently be written as

$$\Pr[J(Y) \neq 1 \mid \Delta_{\text{ch}}] = 1 - \left[F_d\left(1 - \frac{\Delta_{\text{ch}}^2}{2}\right)\right]^{K-1}.$$

Corollary 1 (Balanced desired and undesired regions) *Suppose exactly G of the K anchors have the desired label, including the source anchor a_1, and labels are assigned independently of anchor directions. Conditional on terminal source-cell exit,*

$$\Pr[\text{undesired} \mid J(Y) \neq 1] = \frac{K-G}{K-1},$$

whereas

$$\Pr[\text{desired} \mid J(Y) \neq 1] = \frac{G-1}{K-1}.$$

Therefore,

$$\Pr[\text{desired} \mid \Theta = \theta] = 1 - \frac{K-G}{K-1}\left\{1 - [1 - C_d(\theta)]^{K-1}\right\}.$$

For an equal desired/undesired split, $G = K/2$,

$$\Pr[\text{undesired} \mid J(Y) \neq 1] = \frac{K}{2(K-1)}.$$

Proof: Conditional on source-cell exit, the winning competing anchor is exchangeable among the $K-1$ non-source anchors. Exactly $K-G$ are undesired and $G-1$ are desired. Combining these probabilities with Theorem 1 proves the result. □

E.2 Empirical validation of the random-codebook prediction

We validate the theorem using a controlled random-anchor survival experiment. For every tested codebook size K, anchors are generated by normalizing independent Gaussian vectors in $\mathbb{R}^{16}$, which yields independent uniform directions on $\mathbb{S}^{15}$. The codebook receives an exactly balanced random binary labeling. At one selected diffusion step, an undesired generated latent position is replaced by a randomly selected desired anchor and rescaled to its pre-intervention latent norm. The diffusion model then performs the remaining denoising updates without accessing competing anchors.

For every generated latent position r, we record the injected anchor a_{k_r} and normalized final latent direction Y_r. We compute

$$\Delta_{\mathrm{ch},r} = \|Y_r - a_{k_r}\|_2$$

and evaluate the exact iid-codebook prediction

$$q_r = 1 - \left[F_d\left(1 - \frac{\Delta_{\mathrm{ch},r}^2}{2}\right)\right]^{K-1}.$$

The predicted terminal source-cell exit rate for a condition is the average

$$\widehat{p}_{\mathrm{exit}} = \frac{1}{R}\sum_{r=1}^{R} q_r.$$

We compare this quantity against the observed terminal source-cell exit rate under the fixed secret codebook,

$$\widehat{p}_{\mathrm{exit}}^{\mathrm{emp}} = \frac{1}{R}\sum_{r=1}^{R} \mathbf{1}\{J(Y_r) \neq J(a_{k_r})\}.$$

We separately measure the undesired-label flip rate and the desired-label occupancy rate. Their iid-codebook predictions are

$$\widehat{p}_N^{\mathrm{pred}} = \frac{K-G}{K-1}\widehat{p}_{\mathrm{exit}}, \qquad \widehat{p}_T^{\mathrm{pred}} = 1 - \widehat{p}_N^{\mathrm{pred}}.$$

This comparison has no fitted parameters: the only input to the theoretical prediction is the observed terminal directional displacement of each latent. Confidence intervals are obtained by bootstrapping prompts rather than treating latent positions within the same continuation as independent. We additionally decode the final latent states into text and re-encode the text with the model's VAE. The re-encoded outcomes provide a separate end-to-end test of whether the latent watermark signal survives the continuous-to-discrete decoding channel.

The results are presented in Figures 8 and 9.

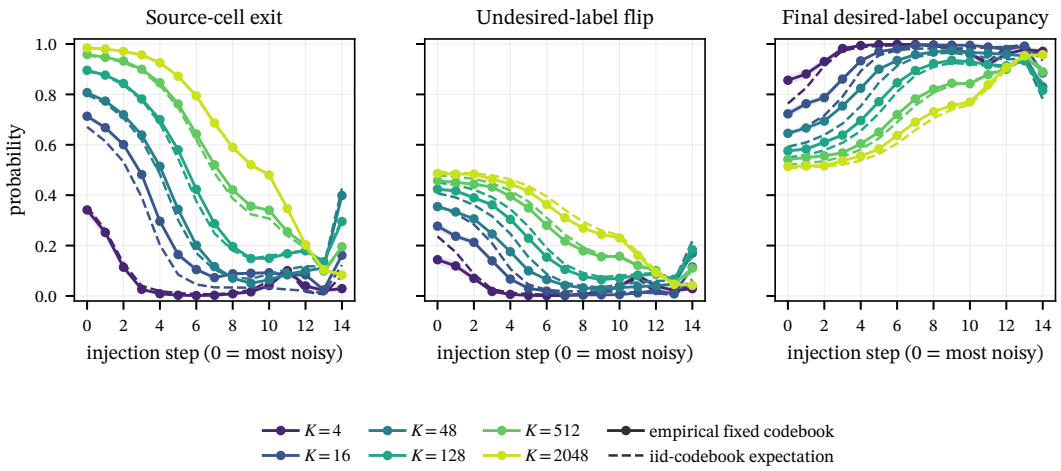


Figure 8: Exact random-anchor survival across watermark insertion steps. Solid curves show empirical outcomes under a fixed secret codebook; dashed curves show the exact iid random-codebook expectation evaluated independently for every measured terminal directional displacement. Shaded regions are prompt-bootstrap 95% confidence intervals. Source-cell exit, undesired-label flip, and final desired-label occupancy are distinct events. The jumps at later steps arise when a position still holds near-pure noise while the model is conditioned on a low noise level, which makes it take larger steps.

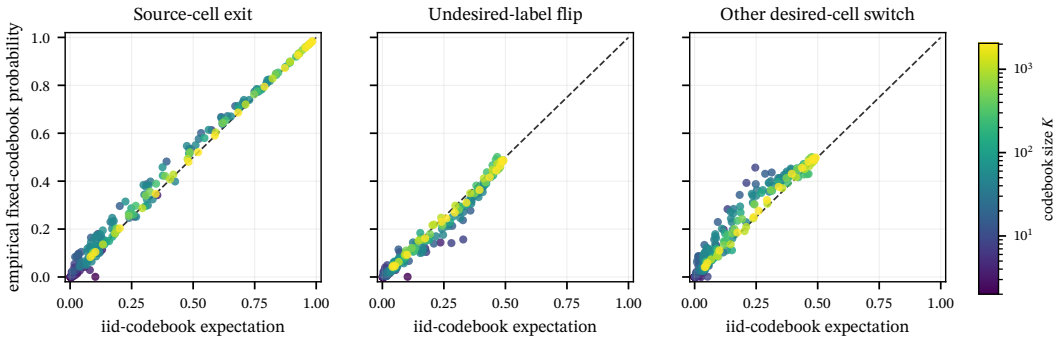


Figure 9: Calibration of the random spherical-codebook prediction. Each point denotes one codebook-size and injection-step condition. The horizontal coordinate is the exact iid-codebook expectation from measured final directional chords; the vertical coordinate is the empirical terminal outcome under a fixed secret codebook. The dashed diagonal denotes exact agreement.

F Hyperparameter Settings

F.1 Baseline Hyperparameters

Published defaults do not always transfer cleanly to CDLMs. For each baseline, we therefore begin with the configuration recommended in the original work and adjust method-specific hyperparameters when needed to obtain a useful balance between detection and text quality.

The image-diffusion baselines additionally require architectural adaptations. The fixed Tree-Ring radius of 10 used in the original image setting is incompatible with the 16×16 latent representation of Cola. We therefore express the radius relative to the latent resolution and evaluate ratios between 0.10 and 0.50. For Tree-Ring and Gaussian Shading, the number of inversion steps matches the generation trajectory: 16 steps for Cola and 64 for ELF-L.

Throughout this section, h denotes the hash-context width and T the sampling temperature. The parameters α_{DiP} and α_{mixing} are specific to DiPmark and TextSeal, respectively, and are unrelated

to the steering strength α of COMPASS. Tables 4 and 5 report the configurations used in the main evaluation.

Table 4: **Baseline configurations for Cola.** Settings used in the main evaluation.

Method	Configuration
Token-level baselines	
KGW	$\gamma = 0.5,\ \delta = 7,\ h = 1,\ T = 1.0$
Unigram	$\gamma = 0.5,\ \delta = 7,\ T = 1.0$
GumbelSoft	$T_{\text{soft}} = 1.5,\ h = 3$
TextSeal	$\alpha_{\text{mixing}} = 0.5,\ h = 3,\ T = 1.4$
DiPmark	$\alpha_{\text{DiP}} = 0.5,\ h = 1,\ T = 1.5$
SynthID	$H = 4,\ m = 30,\ T = 1.5$
Initial-noise baselines	
Tree-Ring	Radius ratios $r \in [0.10, 0.50]$, amplitude $= 5$, 16 inversion steps
Gaussian Shading	16-bit payload per token vector, 16 inversion steps

Table 5: **Baseline configurations for ELF-L.** Settings used in the main evaluation.

Method	Configuration
Token-level baselines	
KGW	$\gamma = 0.5,\ \delta = 18,\ h = 1,\ T = 1.0$
Unigram	$\gamma = 0.5,\ \delta = 19,\ T = 1.0$
GumbelSoft	$T_{\text{soft}} = 0.3,\ h = 3,\ T = 2.8$
TextSeal	$\alpha_{\text{mixing}} = 0.1,\ h = 3,\ T = 2.8$
DiPmark	$\alpha_{\text{DiP}} = 0.50,\ h = 5,\ T = 4.0$
SynthID	$H = 4,\ m = 30,\ T = 3.0$
Initial-noise baselines	
Tree-Ring	Radius ratios $r \in [0.18, 0.71]$, amplitude $= 3$, 64 inversion steps
Gaussian Shading	512-bit payload per token vector, 64 inversion steps

F.2 COMPASS Hyperparameters for Cola

COMPASS has five hyperparameters: the codebook size K, watermarking window $[s_{\text{start}}, s_{\text{end}}]$, selection fraction f, steering strength α, and noise strength σ (Sec. 3). The main Cola results in Table 1 use

$$K = 46, \qquad [s_{\text{start}}, s_{\text{end}}] = [4, 6], \qquad \alpha = 0.96, \qquad \sigma = 0.005, \qquad f = 1.$$

F.3 COMPASS Hyperparameters for ELF-L

The main ELF-L results in Table 2 use

$$K = 20, \qquad [s_{\text{start}}, s_{\text{end}}] = [10, 15], \qquad \alpha = 1, \qquad \sigma = 0.6, \qquad f = 0.2.$$

F.4 Hyperparameter Ranges for the Pareto Plot

The main evaluation reports one configuration per method, whereas the Pareto plot in Figure 3 shows how detection and text quality vary across multiple configurations. Table 6 reports the ranges covered by these configurations. Parameters shown as single values remain fixed throughout the sweep.

Table 6: **Hyperparameter sweep for the Cola Pareto plot.** Intervals denote the ranges covered by the evaluated configurations.

Method	Configuration range
Token-level baselines	
KGW	$\gamma = 0.5,\ \delta \in [2, 12],\ h = 1,\ T = 1.0$
Unigram	$\gamma = 0.5,\ \delta \in [2, 12],\ T = 1.0$
GumbelSoft	$T_{\text{soft}} \in [0.3, 2.0],\ h = 3,\ T = 1.0$
TextSeal	$\alpha_{\text{mixing}} \in \{0.1, 0.5\},\ h = 3,\ T \in [1.0, 2.0]$
DiPmark	$\alpha_{\text{DiP}} = 0.5,\ h = 1,\ T \in [1.0, 2.0]$
SynthID	$H = 4,\ m = 30,\ T \in [1, 2]$
Initial-noise baselines	
Tree-Ring	Radius ratios $r \in [0.10, 0.50]$, amplitude $\in [1, 7]$, 16 inversion steps
Gaussian Shading	16-bit payload per token vector, 16 inversion steps
Proposed method	
COMPASS (ours)	$K \in [40, 46]$, $\alpha \in [0.72, 0.9875]$, $s_{\text{start}} \in [1, 4]$, $s_{\text{end}} \in [3, 7]$, $\sigma \in [0.0015, 0.03]$, $f \in [0.7812, 1.0]$

F.5 Stronger Baseline Configurations for Paraphrasing

A robustness comparison is informative only when each watermark can be detected reliably before the attack. For the DIPPER evaluation, we therefore strengthen the token-level baselines until their unattacked TPRs are comparable to that of COMPASS, and then hold these configurations fixed across all paraphrasing strengths. As shown in Table 3, their detection rates nevertheless decline sharply under paraphrasing. Tables 7 and 8 report the strengthened configurations.

Table 7: **Strengthened baseline configurations for DIPPER paraphrasing on Cola.**

Method	Configuration
KGW	$\gamma = 0.5,\ \delta = 9,\ h = 1,\ T = 1.0$
Unigram	$\gamma = 0.5,\ \delta = 10,\ T = 1.0$
GumbelSoft	$T_{\text{soft}} = 1.875,\ h = 3,\ T = 1.0$
TextSeal	$\alpha_{\text{mixing}} = 0.1,\ h = 3,\ T = 1.5$

Table 8: **Strengthened baseline configurations for DIPPER paraphrasing on ELF-L.**

Method	Configuration
KGW	$\gamma = 0.5,\ \delta = 20,\ h = 1,\ T = 1.0$
Unigram	$\gamma = 0.5,\ \delta = 21,\ T = 1.0$
GumbelSoft	$T_{\text{soft}} = 3,\ h = 3,\ T = 1.0$
TextSeal	$\alpha_{\text{mixing}} = 0.1,\ h = 3,\ T = 3.5$

F.6 Adapting Gaussian Shading to Text Watermarking

We adapt the latent-sign construction of Gaussian Shading (Yang et al., 2024) to continuous text latents. Let $m \in \{0, 1\}^d$ denote the payload, where d is the latent dimension. We mask m with a keyed pad and encode the resulting pattern in the signs of the initial-noise coordinates while preserving their magnitudes. The same d-bit pattern is repeated across token positions rather than sampled independently.

Table 9: **Post-generation attacks.** For deletion, truncation, and swapping, percentages are measured relative to the number of words. For Typo5, the percentage is measured relative to alphabetic characters. For CP25 and CP50, it is the expected fraction of aligned complete sentences replaced with clean text.

Family	Abbreviation	Operation
Deletion and truncation	WD10, WD30, WD50	Delete 10%, 30%, or 50% of words uniformly at random
	Span30	Delete a uniformly positioned contiguous span containing 30% of the words
	First50	Retain only the first 50% of the words
	Last50	Retain only the last 50% of the words
Surface edits	Swap20	Swap non-overlapping adjacent word pairs, affecting up to 20% of the words
	Typo5	Replace 5% of alphabetic characters with randomly selected letters
	Shuffle	Randomly permute the order of complete sentences
Clean-text dilution	CP25, CP50	Replace an expected 25% or 50% of position-aligned complete sentences with the corresponding sentences from the paired clean continuation

At detection time, we pad or truncate the candidate text to the evaluation length, re-encode it, and invert the denoising process from the recovered latent. The detector then reads the reconstructed sign pattern from the inverted noise.

G Post-Generation Attack Protocol

We test whether the watermark survives edits made to the text after it has been decoded. Each attack is a map from strings to strings. It needs no access to the model, the key, or the detector. Table 9 lists the attacks and the column abbreviations used in the robustness tables.

Evaluation texts. For Cola, each text is a continuation of a 128-token prompt, and the model generates 128 new tokens. We evaluate on 600 watermarked continuations. ELF-L is an unconditional sampler and accepts no prompt. We generate texts of 1,024 tokens and split each into eight consecutive, non-overlapping pieces of 128 tokens. Each piece is evaluated as a separate text. All texts therefore contain 128 generated tokens, and every method is attacked and scored on the same texts.

Preprocessing. For Cola, we cut each continuation at its first end-of-text marker and keep at most 128 tokens under the Cola tokenizer. Attacks act on the decoded string. Unless stated otherwise, an attack splits the string on whitespace into words $w_1, \ldots, w_n$, edits this sequence, and joins the result with single spaces.

G.1 Text-Space Attacks

Deletion and truncation. Random word deletion removes $\mathrm{round}(rn)$ words, chosen uniformly without replacement, for $r \in \{0.1, 0.3, 0.5\}$. Span deletion removes a single contiguous run of $k = \mathrm{round}(0.3n)$ words. Its start index is uniform over the $n - k + 1$ valid positions. Truncation keeps either the first or the last $\lceil n/2 \rceil$ words. Span deletion and truncation remove a comparable share of words to random deletion but leave the surviving words contiguous.

Surface edits. Adjacent-word swapping selects at most $\mathrm{round}(0.1n)$ non-overlapping pairs (w_j, w_{j+1}), with start indices j drawn uniformly from the valid positions. It then exchanges the two words of every selected pair, so at most 20% of the words change position. Character typos act on the string directly and do not split it into words. We select $\mathrm{round}(0.05m)$ of the m alphabetic characters uniformly at random. Each selected character is replaced by a letter drawn uniformly from the alphabet, in the case of the original. Whitespace, digits, and punctuation are left unchanged. Sentence shuffling splits the text after every `.`, `!`, or `?` followed by whitespace. It applies a uniformly random permutation to the resulting sentences. A text with fewer than two sentences is left unchanged.

Sentence copy-paste attacks. For Cola, we pair each watermarked continuation with the clean continuation generated from the same prompt and truncate both to 128 tokens. For ELF-L, we split each 1,024-token watermarked and clean generation into eight consecutive, non-overlapping chunks of 128 tokens and pair chunks at the same position. Within each pair, we align complete sentences by order. For CP25 and CP50, we randomly replace approximately 25% or 50% of the watermarked sentences with the corresponding clean sentences. Unmatched sentences and incomplete trailing text remain unchanged.

H Ablation Studies

We ablate the main design choices of COMPASS by varying one parameter at a time while holding the remaining configuration fixed. In each ablation, the control denotes the configuration used for the corresponding model in our main evaluation, as reported in Sec. 4.

Detection. For Cola, we generate 128-token continuations from a fixed set of 600 prompts. Each condition produces 600 watermarked continuations, which we compare with the same 600 clean continuations using the standardized mean-margin detector. We follow the corresponding model-specific protocol for ELF-L. We report the true-positive rates at 1% and 5% false-positive rates (TPR@1% and TPR@5%), together with the area under the receiver operating characteristic curve (AUROC).

Generation quality. We use the two complementary quality measures from Appx. K. CompassJudger-2-7B evaluates fluency, coherence, repetition, naturalness, and meaningfulness on a degradation scale from 0 (no degradation) to 4 (severe degradation). We report the relative change in the mean score:

$$\Delta_{\mathrm{deg}} = 100\,\frac{\bar{d}_{\mathrm{marked}} - \bar{d}_{\mathrm{clean}}}{\bar{d}_{\mathrm{clean}}}.$$

For Cola, the clean reference has $\bar{d}_{\mathrm{clean}} = 2.105$, while the control condition has $\bar{d}_{\mathrm{marked}} = 2.296$, yielding $\Delta_{\mathrm{deg}} = +9.07\%$. For ELF-L, we apply the same definition using its model-specific clean reference.

Our second measure is the absolute log-perplexity deviation:

$$|\Delta \log \mathrm{PPL}| = |\log \mathrm{PPL}_{\mathrm{marked}} - \log \mathrm{PPL}_{\mathrm{clean}}|\,.$$

Taking the absolute value treats increases and decreases in perplexity symmetrically and measures the magnitude of the deviation from the clean output.

Reading the tables. Arrows indicate whether higher or lower values are preferred. Boldface marks the best value in each column, and underlining marks the second-best value; ties receive the same marking. The control configuration is labeled explicitly.

Summary. The two models exhibit different sensitivity profiles. For Cola, steering strength and watermark-window placement produce the clearest detection–quality trade-offs. Codebook size and intervention fraction also affect detection, whereas the tested noise scales have comparatively little influence. Detection on ELF-L saturates across all tested windows and intervention fractions, but weak steering and very large codebooks reduce TPR. Once detection saturates, the quality metrics become the main criterion for distinguishing configurations.

H.1 Codebook Size

The codebook size K determines the number of anchor directions available in the target and non-target sets. Its effect on Cola is not monotonic (Table 10). The control value $K = 46$ achieves the highest TPR@1% and the lowest judged degradation. Increasing the codebook to $K = 64$ and $K = 96$ reduces TPR@1% by more than 26 and 33 percentage points, respectively.

On ELF-L, detection is perfect for $K \in \{16, 20, 40\}$, but declines for substantially larger codebooks, reaching 98.8% at $K = 256$ and 91.7% at $K = 512$ (Table 11). Thus, although ELF-L is less sensitive to moderate changes in K, excessively large codebooks still weaken detection.

Table 10: **Effect of codebook size K on Cola.**

K	TPR@1% ↑	TPR@5% ↑	AUROC ↑	Deg. (%) ↓	$\lvert\Delta \log \mathrm{PPL}\rvert$ ↓
24	88.50	96.83	0.9892	+14.55	1.13
32	92.50	**98.00**	**0.9939**	+17.34	**0.94**
42	86.67	96.50	0.9909	+11.29	1.22
46 (control)	**94.33**	97.67	0.9934	**+9.07**	1.09
64	68.00	93.50	0.9821	+10.66	1.35
96	61.33	88.17	0.9725	+9.56	1.31

Table 11: **Effect of codebook size K on ELF-L.**

K	TPR@1% ↑	TPR@5% ↑	AUROC ↑	Deg. (%) ↓	$\lvert\Delta \log \mathrm{PPL}\rvert$ ↓
2	99.7	**100.00**	0.9999	+26.11	**0.126**
16	**100.00**	**100.00**	0.9999	+17.61	0.896
20 (control)	**100.00**	**100.00**	**1.0000**	**+4.33**	0.623
40	**100.00**	**100.00**	**1.0000**	+17.63	1.238
256	98.8	99.5	0.999	+15.32	2.057
512	91.7	98.2	0.993	+16.00	2.098

H.2 Noise Scale

The noise term separates representations that would otherwise be steered toward the same anchor direction. Within the tested range, its scale has little effect on Cola (Table 12): TPR@1% remains between 93.17% and 96.17%, judged degradation ranges from $+9.07\%$ to $+12.40\%$, and $|\Delta \log \mathrm{PPL}|$ remains nearly constant.

Detection on ELF-L remains perfect through $\sigma = 1$ and declines only slightly, to 99.00% TPR@1%, at $\sigma = 2$ (Table 13). The quality metrics vary more noticeably, indicating that additional noise can affect the generated text even when detection has already saturated.

Table 12: **Effect of noise scale σ on Cola.**

σ	TPR@1% ↑	TPR@5% ↑	AUROC ↑	Deg. (%) ↓	$\lvert\Delta \log \mathrm{PPL}\rvert$ ↓
0.0	94.67	98.50	0.9953	+11.27	1.13
0.0025	**96.17**	**98.67**	**0.9956**	+11.24	1.13
0.005 (control)	94.33	97.67	0.9934	**+9.07**	**1.09**
0.01	93.17	97.00	0.9947	+12.40	**1.09**
0.03	95.17	98.33	0.9947	+10.23	**1.09**
0.06	93.17	97.33	0.9947	+11.59	**1.09**

Table 13: **Effect of noise scale σ on ELF-L.**

σ	TPR@1% ↑	TPR@5% ↑	AUROC ↑	Deg. (%) ↓	$\lvert\Delta \log \mathrm{PPL}\rvert$ ↓
0.2	**100.00**	**100.00**	**1.0000**	+16.61	1.148
0.4	**100.00**	**100.00**	**1.0000**	+16.13	1.096
0.6 (control)	**100.00**	**100.00**	**1.0000**	**+4.33**	0.623
1	**100.00**	**100.00**	**1.0000**	+5.13	0.738
2	99.00	99.83	0.9996	+9.49	**0.128**

H.3 STEERING STRENGTH

The steering strength α determines how far each selected representation moves toward its target anchor. Together with watermark-window placement, it produces the clearest detection–quality trade-off among the parameters considered.

On Cola, weak steering at $\alpha = 0.5$ causes almost no measured degradation but yields only 7.00% TPR@1% (Table 14). Detection rises steadily with α, reaching 74.00% at $\alpha = 0.8$ and at least 89.50% for $\alpha \geq 0.9$, at the cost of greater quality deviation.

The transition is sharper on ELF-L (Table 15). TPR@1% rises from 3.00% at $\alpha = 0.05$ to 16.33% at $\alpha = 0.2$, then reaches 97.33% at $\alpha = 0.4$ and 100.00% for $\alpha \geq 0.5$. The control configuration retains perfect detection while substantially reducing judged degradation relative to $\alpha = 0.5$.

Table 14: **Effect of SLERP steering strength α on Cola.**

α	TPR@1% ↑	TPR@5% ↑	AUROC ↑	Deg. (%) ↓	$\lvert\Delta \log \mathrm{PPL}\rvert$ ↓
0.5	7.00	16.17	0.7165	**+0.22**	**0.17**
0.65	31.33	51.17	0.8992	+1.66	0.45
0.8	74.00	87.33	0.9771	+7.14	0.79
0.9	89.50	95.50	0.9921	+10.47	1.04
0.96 (control)	94.33	**97.67**	0.9934	+9.07	1.09
1.0	**95.00**	97.33	**0.9957**	+13.90	1.17

Table 15: **Effect of SLERP steering strength α on ELF-L.**

α	TPR@1% ↑	TPR@5% ↑	AUROC ↑	Deg. (%) ↓	$\lvert\Delta \log \mathrm{PPL}\rvert$ ↓
0.05	3.00	8.50	0.5319	**-0.09**	**0.009**
0.2	16.33	28.00	0.7356	+0.29	0.063
0.4	97.33	99.17	0.9983	+16.84	0.055
0.5	**100.00**	**100.00**	**1.0000**	+17.41	0.457
1 (control)	**100.00**	**100.00**	**1.0000**	+4.33	0.623

H.4 INTERVENTION FRACTION

The intervention fraction f specifies the share of eligible positions steered at each watermarking step. On Cola, TPR@1% rises from 57.00% at $f = 0.25$ to 95.17% at $f = 0.6$, after which it plateaus (Table 16). The control setting $f = 1$ remains within one percentage point of the maximum while producing less judged degradation than $f = 0.6$. On ELF-L, every tested fraction achieves perfect detection, so the quality metrics determine the preferred operating point (Table 17).

Table 16: **Effect of intervention fraction f on Cola.**

f	TPR@1% ↑	TPR@5% ↑	AUROC ↑	Deg. (%) ↓	$\lvert\Delta \log \mathrm{PPL}\rvert$ ↓
0.25	57.00	72.67	0.9533	**+7.13**	**0.98**
0.4	83.83	92.00	0.9862	+10.51	1.19
0.6	**95.17**	**97.83**	**0.9951**	+13.03	1.26
0.8	93.00	97.17	0.9939	+11.70	1.11
0.9688	94.17	96.83	0.9933	+9.12	1.08
1 (control)	94.33	97.67	0.9934	+9.07	1.09

Table 17: **Effect of intervention fraction f on ELF-L.**

f	TPR@1% ↑	TPR@5% ↑	AUROC ↑	Deg. (%) ↓	$\lvert\Delta \log \mathrm{PPL}\rvert$ ↓
0.1	**100.00**	**100.00**	**1.0000**	+6.94	0.80
0.2 (control)	**100.00**	**100.00**	**1.0000**	+4.33	**0.62**
0.6	**100.00**	**100.00**	**1.0000**	**+3.84**	1.27
1	**100.00**	**100.00**	**1.0000**	+3.89	1.26

H.5 Watermarking Window

For Cola, we vary the start, end, and length of the watermarking window in three separate sweeps. Let $L = s_{\mathrm{end}} - s_{\mathrm{start}} + 1$ denote the inclusive length of the window. The start-step sweep fixes $s_{\mathrm{end}} = 6$ (Table 18), whereas the end-step sweep fixes $s_{\mathrm{start}} = 4$ (Table 19). The length sweep uses a predetermined pair $(s_{\mathrm{start}}, s_{\mathrm{end}})$ for each L (Table 20). Because these intervals share neither a common center nor a common boundary, the final sweep compares the specified windows rather than isolating length independently of position.

Later intervention strengthens detection on Cola but carries a substantial quality cost. Extending the control window from $[4, 6]$ to $[4, 9]$, for example, raises TPR@1% from 94.33% to 99.83%, while increasing judged degradation from $+9.07\%$ to $+58.62\%$. Earlier windows preserve quality more effectively but leave a weaker signal. The control window lies between these regimes.

Every tested ELF-L window achieves perfect detection (Table 21). Moving the window later generally reduces judged degradation, from $+5.113\%$ for $[5, 10]$ to $+2.457\%$ for $[25, 35]$, while increasing $|\Delta \log \mathrm{PPL}|$ from 0.9821 to 1.716. The opposing trends confirm that the two quality measures capture different properties of the generated text.

Table 18: **Effect of the watermark-window start s_{start} on Cola.** The end is fixed at $s_{\mathrm{end}} = 6$.

s_{start} (window)	TPR@1% ↑	TPR@5% ↑	AUROC ↑	Deg. (%) ↓	$\lvert\Delta \log \mathrm{PPL}\rvert$ ↓
1 (1–6)	78.83	90.17	0.9854	+7.78	**0.67**
2 (2–6)	83.83	93.50	0.9886	+8.14	0.72
3 (3–6)	87.17	95.50	0.9895	**+7.13**	0.85
4 (4–6; control)	94.33	97.67	0.9934	+9.07	1.09
5 (5–6)	99.33	99.67	0.9994	+20.57	1.51
6 (6–6)	**99.83**	**100.00**	**0.9999**	+42.77	1.93

Table 19: **Effect of the watermark-window end s_{end} on Cola.** The start is fixed at $s_{\mathrm{start}} = 4$.

s_{end} (window)	TPR@1% ↑	TPR@5% ↑	AUROC ↑	Deg. (%) ↓	$\lvert\Delta \log \mathrm{PPL}\rvert$ ↓
4 (4–4)	70.17	85.17	0.9768	**+4.10**	**0.75**
5 (4–5)	83.33	93.67	0.9897	+7.06	0.89
6 (4–6; control)	94.33	97.67	0.9934	+9.07	1.09
7 (4–7)	98.33	99.33	0.9987	+21.14	1.39
8 (4–8)	99.50	**99.83**	0.9997	+34.77	1.75
9 (4–9)	**99.83**	**99.83**	**0.9998**	+58.62	2.01

Table 20: **Effect of watermark-window length L on Cola.** Parentheses give $(s_{\text{start}}, s_{\text{end}})$ for each length.

L (window)	TPR@1% ↑	TPR@5% ↑	AUROC ↑	Deg. (%) ↓	$\lvert\Delta \log \text{PPL}\rvert$ ↓
1 (5–5)	95.67	98.17	0.9961	+16.42	1.34
2 (4–5)	83.33	93.67	0.9897	**+7.06**	0.89
3 (4–6; control)	94.33	97.67	0.9934	+9.07	1.09
4 (3–6)	87.17	95.50	0.9895	+7.13	**0.85**
5 (3–7)	**96.50**	**98.67**	**0.9971**	+15.80	1.21
6 (2–7)	95.33	98.33	**0.9971**	+15.55	1.12

Table 21: **Effect of the watermarking window $[s_{\text{start}}, s_{\text{end}}]$ on ELF-L.**

$[s_{\text{start}}, s_{\text{end}}]$	TPR@1% ↑	TPR@5% ↑	AUROC ↑	Deg. (%) ↓	$\lvert\Delta \log \text{PPL}\rvert$ ↓
5–10	**100.00**	**100.00**	**1.0000**	+5.11	0.98
10–15 (control)	**100.00**	**100.00**	**1.0000**	+4.33	**0.62**
10–20	**100.00**	**100.00**	**1.0000**	+4.63	1.21
20–30	**100.00**	**100.00**	**1.0000**	+3.08	1.51
25–35	**100.00**	**100.00**	**1.0000**	**+2.46**	1.72

I Computational Cost

We measure the computational cost of COMPASS on a single NVIDIA A40 with batch size one. Each result uses 100 paired unwatermarked and watermarked runs. The members of each pair share the prompt for Cola and the initial-noise and denoising seeds for ELF-L. The Cola runs use a 128-token prompt, generate 128 tokens, and perform 16 denoising steps. The ELF-L runs generate 1,024 tokens unconditionally using the official 64-step SDE sampler. Timing begins before latent initialization and ends when the decoded text is returned. It includes denoising, watermark injection, and decoding, but excludes model loading, data access, serialization, and detection. Codebooks are prepared and cached before timing.

Table 22: **Computational cost of COMPASS.** Latencies are medians over 100 runs. The value before brackets is the estimated average latency overhead; brackets give its 95% confidence range across matched unwatermarked and watermarked runs.

Model	Method	Latency (s)	Overhead (%)	Tokens/s	Peak memory (GB)
Cola	Unwatermarked	12.873	–	9.91	12.29
Cola	COMPASS	12.919	$+0.34$ $[+0.21, +0.54]$	9.90	12.29
ELF-L	Unwatermarked	3.302	–	309.9	2.86
ELF-L	COMPASS	3.358	$+1.63$ $[+1.47, +1.79]$	304.8	2.86

As Table 22 shows, COMPASS increases end-to-end generation latency by only 0.34% on Cola and 1.63% on ELF-L, without increasing peak accelerator memory. When profiled separately and summed over all active watermarking steps, the injection operations take 60.3 ms for Cola and 53.3 ms for ELF-L. These figures are conservative upper bounds because the profiling procedure synchronizes the accelerator after every measured phase.

Codebook creation is a one-time cost paid before generation starts, so it adds nothing to the generation loop. For Cola, creating the Gaussian codebook takes 0.20 ms. For ELF-L, the full setup takes 149 ms. Most of this time goes to reading the saved corpus data from disk and verifying it. Building the codebook itself ($K = 20$ anchors, 512 dimensions) takes only 26.7 ms, or 31.2 ms if sampling runs on the CPU instead of the GPU. Once the codebook exists, each lookup during generation takes just 0.014 ms.

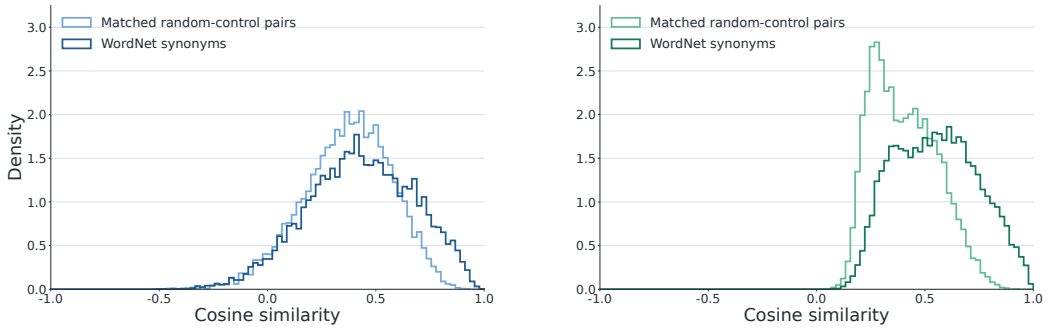
Detection is measured separately from generation and is similarly cheap: it takes 8.26 ms for a 128-token Cola continuation and 14.45 ms for a 1,022-token ELF-L text.

J Semantic Similarity in Cola and ELF-L

COMPASS assigns encoder representations to anchors using cosine similarity. To see whether this geometry reflects word meaning, we compare the cosine similarity of WordNet (Miller, 1995) synonym pairs with that of random control pairs. We use 11,027 WordNet pairs, each containing two synonymous words. To construct the corresponding control, we keep one word fixed and replace the other with a randomly selected word matched for part of speech and tokenizer length. We encode every word on its own, without surrounding context. For Cola, we take token representations from the VAE encoder; for ELF-L, we use its T5-small semantic encoder. When a word spans several token positions, we average the corresponding representations and normalize the result.

For Cola, the synonym and random-control distributions overlap heavily (Figure 10). Their mean cosine similarities are 0.428 and 0.368, respectively. The separation is considerably clearer for ELF-L, whose synonym scores are shifted further toward higher similarities. Because the models use different encoders, their cosine scores are not directly comparable; we therefore compare synonyms with controls separately within each model.

This pattern is consistent with the paraphrasing results in Table 3. During detection, COMPASS compares each encoded word representation with the target and non-target anchors. If a paraphrase replaces a word with a synonym that has a similar representation, these anchor similarities, and therefore the word's contribution to the detection score, should change only slightly. The stronger similarity between synonyms in ELF-L may thus help its watermark survive paraphrasing.



(a) Cola VAE

(b) ELF-L T5-small semantic encoder

Figure 10: **Cosine similarity of WordNet synonym pairs and matched random-control pairs.** The distributions contain 11,027 synonym pairs and their corresponding random controls, with every word encoded without surrounding context. The distributions overlap heavily for Cola, whereas ELF-L shows a clearer shift toward higher similarity for synonyms.

K Text Generation Metrics

We evaluate text quality using two complementary measures. Perplexity quantifies how likely a generated sequence is under a fixed reference language model. An LLM judge scores aspects of quality that likelihood does not directly measure, including coherence and repetition.

K.1 Perplexity

Let $x = (x_1, \ldots, x_T)$ be the token sequence of a generated text under the reference model's tokenizer, and let c denote any preceding context. We compute perplexity over the generated tokens as

$$\mathrm{PPL}_\theta(x \mid c) = \exp\left(-\frac{1}{T}\sum_{t=1}^{T}\log p_\theta(x_t \mid c, x_{<t})\right). \tag{3}$$

In Eq. (3), only x contributes to the average; c conditions the probabilities. Lower perplexity indicates higher average likelihood under the reference model. We use the same model and tokenizer to score every method.

The reference model is the pretrained, non-instruction-tuned `meta-llama/Llama-3.1-8B` checkpoint. For ELF-L, c is empty. We compute $\Delta \log \mathrm{PPL}$ by subtracting the clean log perplexity from the watermarked log perplexity. Positive values therefore indicate higher perplexity for watermarked text under the reference model.

K.2 LLM JUDGE

We also evaluate each generation with `opencompass/CompassJudger-2-7B-Instruct`. The judge assigns separate scores for fluency, coherence, repetition, naturalness, and meaningfulness on a degradation scale from 0 (excellent) to 4 (severe degradation). Lower scores indicate better quality.

For Cola, the judge receives the source text and generated continuation (Figure 11). For ELF-L, it receives only the generated text (Figure 12). The figures reproduce the prompts verbatim; fields in braces are populated at evaluation time. The ELF-L prompt instructs the judge to evaluate the completion as a standalone text, with no source prompt or reference answer.

System Prompt Cola, conditional continuation

```
Please act as an impartial judge to evaluate the linguistic quality of a
generated text completion given the preceding text. Your evaluation should
focus on the following criteria: fluency, coherence, repetition, naturalness,
and meaningfulness.

The Source Text is the text that precedes the generated completion. The
Generated Completion should be treated as a continuation of the Source Text,
not as an answer to a question or an assistant response. Evaluate each
criterion independently using the following degradation scale:

0 = Excellent. No meaningful quality problems are present.
1 = Minor degradation. Small imperfections are present, but the text remains
  highly natural and readable.
2 = Moderate degradation. Noticeable quality problems are present, but the
  text remains generally understandable.
3 = Significant degradation. Clear and frequent problems noticeably reduce
  text quality.
4 = Severe degradation. The text is substantially broken, incoherent, highly
  repetitive, nonsensical, fragmented, or otherwise very poor.

Lower scores indicate better quality.

Use the following criteria:

Fluency: grammatical correctness, readability, sentence structure, and
  smoothness of the generated text.
Coherence: how logically and consistently the completion continues the Source
  Text and develops its ideas.
Repetition: unnecessary repetition or redundancy of words, phrases, sentences,
  or ideas.
Naturalness: how natural and well-formed the completion sounds as text,
  without awkward, mechanical, or implausible phrasing.
Meaningfulness: whether the completion communicates understandable and
  meaningful content that appropriately continues the Source Text.

Important rules:

Evaluate only the Generated Completion. Use the Source Text only as context.
Do not treat the Source Text as a question or instruction. Do not evaluate
helpfulness, instruction following, or question answering. Do not perform
detailed fact-checking. Do not compare the completion with another completion.
Do not let completion length influence your judgment unless it directly
causes a quality problem. Do not assume whether the text was generated with
or without a watermark, and do not attempt to identify any watermarking
method. Do not invent minor problems merely to avoid assigning a score of 0.
Assign each criterion independently.

Your final reply must be structured in the following format:

{
   "Fluency": "[0, 1, 2, 3, or 4]",
   "Coherence": "[0, 1, 2, 3, or 4]",
   "Repetition": "[0, 1, 2, 3, or 4]",
   "Naturalness": "[0, 1, 2, 3, or 4]",
   "Meaningfulness": "[0, 1, 2, 3, or 4]"
}

Source Text: {source_text}

Generated Completion: {completion}

Now it's your turn. Please provide the evaluation result as required:
```

Figure 11: Prompt used to judge Cola generations. The source text and generated continuation are both supplied.

System Prompt ELF-L, unconditional generation

```
Please act as an impartial judge to evaluate the linguistic quality of a
generated text completion. The completion is an unconditional ELF-L text
sample and must be evaluated as a standalone text. There is no source prompt
or reference answer. Evaluate only the Generated Completion.

Use the following degradation scale independently for each criterion:

0 = Excellent. No meaningful quality problems are present.
1 = Minor degradation. Small imperfections are present, but the text remains
  highly natural and readable.
2 = Moderate degradation. Noticeable quality problems are present, but the
  text remains generally understandable.
3 = Significant degradation. Clear and frequent problems noticeably reduce
  text quality.
4 = Severe degradation. The text is substantially broken, incoherent, highly
  repetitive, nonsensical, fragmented, or otherwise very poor.

Lower scores indicate better quality.

Fluency: grammatical correctness, readability, sentence structure, and
  smoothness.
Coherence: logical consistency and whether the text develops a comprehensible
  passage.
Repetition: unnecessary repetition or redundancy of words, phrases, sentences,
  or ideas.
Naturalness: how natural and well-formed the text sounds, without awkward or
  mechanical phrasing.
Meaningfulness: whether the text communicates understandable and meaningful
  content.

Do not perform detailed fact-checking. Do not compare this text with another
text. Do not let text length influence your judgment unless it directly
causes a quality problem. Do not attempt to identify watermarking or infer
how the text was generated.

Your final reply must be exactly one JSON object with these five integer
fields:

{"Fluency": "[0, 1, 2, 3, or 4]", "Coherence": "[0, 1, 2, 3, or 4]",
"Repetition": "[0, 1, 2, 3, or 4]", "Naturalness": "[0, 1, 2, 3, or 4]",
"Meaningfulness": "[0, 1, 2, 3, or 4]"}

Generated Completion: {completion}

Now provide the evaluation result as required:
```

Figure 12: Prompt used to judge ELF-L generations. The completion is evaluated as a standalone text without a source prompt or reference answer.

L Qualitative Comparison of Watermarked and Unwatermarked Generations

To complement the aggregate generation-quality results, we present unwatermarked and watermarked generations from Cola and ELF-L. For Cola, each pair contains continuations produced from the same prompt. Because ELF-L generates text unconditionally, its examples contain no prompt: each pair consists of an unwatermarked and a watermarked 1,024-token generation sampled from the same initial noise, of which we show the same 128-token segment. All text is reproduced verbatim, including grammatical errors, repetitions, and truncated endings.

The examples are generated with the following COMPASS settings. For Cola, we use $K = 46$ anchors, the watermarking window $[4, 6]$, steering strength $\alpha = 0.96$, noise strength $\sigma = 0.005$, and selection fraction $f = 1$. For ELF-L, we use $K = 20$ anchors, the watermarking window $[10, 15]$, steering strength $\alpha = 0.35$, noise strength $\sigma = 0.9$, and selection fraction $f = 0.2$. The Cola setting matches the configuration reported in Table 1. The ELF-L examples were generated using a weaker steering strength and a larger noise scale than the main configuration reported in Table 2 ($\alpha = 1$, $\sigma = 0.6$). Accordingly, they illustrate the qualitative effect of watermarking rather than the outputs produced at the evaluated operating point.

L.1 COLA

Cola generation example 1

Prompt

Farmers are set to be hit with a significant increase in fertiliser prices next year, according to researchers at Teagasc. It has said that given the developments in global supply and global, the outlook for international fertiliser prices in 2019 is for prices for most fertilisers to increase relative to 2018 levels. In its outlook for 2019 it said that fertiliser prices in 2019 are forecast to be on average 16pc higher than in 2018. Dr. Fiona Thorne, Senior Research Officer with Teagasc said there are a number of factors which are influencing its forecast. “On the

Unwatermarked continuation

one hand, there is the ‘ significant growth in global supply of fertilisers and the increases in demand in a few countries’” she said. On the other hand, there are the developments in the market environment in which fertiliser prices could go up. She said, inevitably, the risk of a rise in prices has to be offset against by the significant average growth for fertilisers as well as in the few and size of farmers of fertilisers. The outlook for international fertiliser prices on 2019 is for prices for most fertilisers to increase relative to 2018 levels. She added that developments in the global supply and global market

Watermarked continuation

one hand we have global developments in supply through the wider market and on the other hand we see a increasing level of market share through the wider market, which have an impact on the up in the fertiliser prices,” she told. Mrs. Thorne said that whilst the increases in fertiliser prices is expected to rise to 16pc in 2019, a number of other factors such as global market developments to climate change could also impact on the price. She continued cited “1 of those factors with a global which could be causing the rise in cost of fertiliser as an additional price rise.” She said that global developments change has a

Cola generation example 2

Prompt

Headphone 'burn-in': Fact or fantasy? A lot of audiophiles believe headphones sound better after a few weeks of use than they do when they're brand-new. Most of my audiophile friends believe that headphones (and speakers and electronics) sound better after the first 100 hours of use than they do when they're brand-new. When I'm doing high-end product reviews I leave the "burn-in, break-in" question up to the manufacturer. If the company's reps claim their product won't sound its best until it has a solid month of use, I'll request a unit with enough hours on it that I can

Unwatermarked continuation

hear it better. If the reps claim their product will take some work when it's new, I'll request a unit with enough hours so that I can hear it better. If a product sounds better after a month has come, I'm more likely to buy it. If an product sounds better after three hours, I'm less likely than buy it. It's a "hICA" for the manufacturer, but they'll try to present the product they know.

I'm interested in what the manufacturer thinks of headphones, and I'm curious if you agree with me. If not, why do you say that is?

Small head (comp

Watermarked continuation

hear it better. If the machine then remains its best, I'll request a larger one(s). If the product says it won't stay its best when it has a solid month of use, I'll request a bigger or larger one. If it takes a product to stay its best without any bad sound until it's available in the marketplace it should request a larger one.

I've ever used headphones for their first 50 hours of use, and as one of my product friends told me, "They don't stay like new." I do believe they sound better than a solid month of use than better units without their high quality because they

Cola generation example 3

Prompt

Coastal markets may cool, and interest rates may increase slightly in the coming year. Priced out of expensive city centers, some buyers desire amenity-rich suburbs instead. By Susan Johnston Taylor, Contributor | Dec. 1, 2015, at 9:00 a.m. At the close of last year, many real estate experts predicted the U.S. Federal Reserve would raise interest rates in 2015. That prediction never came to be, but even if rates rise later this month or next year, Ralph McLaughlin, a housing economist at Trulia, doesn't think it will scare away many buyers.

Unwatermarked continuation

"Ultimately, I think rates are down," McLaughlin said, but the question of whether that rise means rates are "lower is one way to look at it." The question, if rates aren't down under McLaughlin's view, is whether the interest market is getting people to buy or rent instead. But the answers are not that simple. If there is no rise in rates, buyers are likely more move on risk-rich places, and if homes have a cheaper real estate price, they may struggle to stay. McLaughlin noted that the first two months from 2014 were "very low in interest home prices, and

Watermarked continuation

"I think a lot of people will still transfer and get their rates down," McLaughlin said. "I think some individuals will go into debt and home." In fear this of those lower buyers who converted their rates earlier this year will be forced to recircuit full interest payments for homes or transfer mortgage arbitrage because the alternative mortgage rates rise as if get a new market rate. "We have seen those type of individuals and the price points they have had to move but their payments don't actually change," McLaughlin said. That risk has offset the housing cost boost for millions of buyers in a slow market, especially high-end areas of the

Cola generation example 4

Prompt

Say you're thinking of launching a side business to make some extra income. Or perhaps you're sick of your day job, and you're ready to put your entrepreneurial skills to the test and turn a passion project into a lucrative profession. Before you build a road map for your business, you need an idea that's both popular and profitable. To help you jump-start your search and get started, use these pro tips and discover small business ideas well-suited for a variety of aspiring entrepreneurs. — Brainstorm, based on your passions and skills. — Determine whether your idea will match your aspirations and meet your goals. Depending on your interests,

Unwatermarked continuation

you may want to start researching small business ideas or work out a few to start with. Once you find out if you know an idea, you can work with a friend or a consultant to discover and create a business idea. Prepare your idea. If you're going to be creating and researching a business idea, plan a schedule and a list of questions to ask about your idea. What are your goals: what do you want to achieve? Where or deadlines do you want to build your business? are the possible products you want to achieve with your business idea. is are some of your deadlines to fulfill and who do you plan to grow

Watermarked continuation

you may brainstorm to more than your business ideas for an idea to create. Determine if your passions and idea meet those your goals you want to grow your business. If not, make sure you find a Carlson to interview and brainstorm business ideas. When asking your staff to support the idea, seek personal permission for your potential of if they agree to speak to it idea after evaluation through an internal survey. — Present your passion that your idea and conduct a research.

Reach your core ideas with a list of witnesses before find two witnesses that share their passions and support their business idea. Perform a research for your way if interviewing these witnesses to speak those your

L.2 ELF-L

ELF-L generation example 1

Unwatermarked generation

s should stand out as a key member of the team's left-handed rotation. "Tayson's part of my staff at 76ers, so we're excited to invite him to the Super Bowl. I'm excited to give him a chance and help him develop throughout the season," said Smith of Torres, who played 28 games over three seasons with New York City. "Over the last few years I've been working on my recovery streak and recovery, knowing that it's the right time for me. I'm ready to make the jump from there as soon as possible

Watermarked generation

to hang longer and be a greater part of the team's left-field lineup. "I don't focus on it. I've been spending a lot of the summer trying to fix it and trying to improve how I'm doing. There's a lot of it to do with pitching," Nichols said after having a broken arm prior to Miami's team's Double-A elimination last year. "I have fractured my shoulders in the past. I don't know if that will happen. I know I'll do it, but that doesn't happen

ELF-L generation example 2

Unwatermarked generation

. We want content to be available on PS4 and Xbox One only, so we can improve the way the game is played. We also want to have a live demo up and running, so that fans can be exposed to some of best games in the series. Those are vital elements to being able to get good reviews on PC. That's what we're looking at for a game that we think will get good reviews and will be available on a mobile device. AP: You're also exploring multiplayer elements for WPG. Will World of Warcraft start picking up on mobile? Is there

Watermarked generation

So the physical graphics we can work on will have more of an impact. You can add maps more as the game goes on. So there is going to be more refinement on the landscape. As we get more into it, the easier it gets. It's a map you see at the beginning of a game, and it's more detailed. We've been working on a gameplay feature that will take more time, and so adding more cities is going to take quite a bit. It needs to be more detailed from a playtesting standpoint. If, say, you're looking at an

ELF-L generation example 3

Unwatermarked generation

. Individual communities and a larger proportion of local governments, businesses and employees can expect to use 1.8 percent less energy over the next decade, according to a End the Debate report, which estimated that the U.S. may be using more than 40 percent less energy by 2050. That figure may leave the U.S. on the cusp of the carbon crisis, at a time when its energy use is growing reduced at an average rate of 25 percent per year. More than 12 million Americans rely on solar panels every day, generating 80 to 90 percent of their total electricity needs from fossil fuel sources

Watermarked generation

. The world as a whole can afford to import about half a trillion barrels worth a day of oil, the IEA estimated. Story Continued Below But despite the slow pace of the financial crisis following the 2008 crisis, other countries will use more of their resources to accelerate oil production. Over the next decade, China will account for 19 percent of global GDP, according to the IEA's Data on Supply and Analysis. Experts estimate China's citizens consume more than 12 million litres worth of oil a month. About 30 to 40 percent of worldwide energy is generated from fossil fuel sources

ELF-L generation example 4

Unwatermarked generation

"It's been very positive, especially from a first-team position," he said. "So many new guys coming into our positions. If we start getting successful on Saturday, of course we'll have to play outside side, but it's something I'm confident that our team has worked hard to achieve." Graham expects significant coverage improvements as he begins to prepare for a tough season. "The guys at our position have experienced it a little bit over the first-team – and obviously over the three weeks we've gotten used to it," he said. The Packers'

Watermarked generation

"It's been very good and we'll have to keep it growing," Halver said. "We need to keep investing in our games. If we are not ready to go. If we're not ready to go, that's fine, honestly. We've got to be more focused on winning our games. So it's been a good year. It's a good year. Over the past couple of years, we did a whole lot of different things. We're going to keep it. We're going to continue to build on it." The more the Packers'

M AI USE STATEMENT

We used generative AI tools to help implement and debug the code for our method and experiments, and to assist in formulating and checking the mathematical proofs and derivations. We also used them to improve the readability of the manuscript, refine plots, and enhance some of the figures. They were not used to generate research ideas, design the method or experiments, produce data, or interpret results; translation, dataset cleaning, and qualitative data analysis are not applicable to this work. The authors verified all proofs and derivations by hand and reviewed and tested all AI-assisted code; every reported number was obtained by running experiments.

N REPRODUCIBILITY STATEMENT

We provide an anonymous code repository at `https://anonymous.4open.science/r/compass-0877/README.md`. It contains the implementations needed to reproduce our reported results for COMPASS and all evaluated baselines on both Cola and ELF-L.