DOI 10.15276/imms.v15.no4.475
UDC 004.056.55:004.8

Informatics and Mathematical Methods in Simulation
Vol.15 (2025), No. 4, pp. 475-485

# LEVERAGING GENERATIVE MODELS FOR CRYPTANALYSIS: EXPLORING THE POTENTIAL OF LLMS AND DIFFUSION MODELS

A. S. Koliada, L. V. Bovnehra

National Odesa Polytechnic University
1, Shevchenko Ave., Odesa, 65044, Ukraine
Email: a.s.koliada@op.edu.ua, dlv5@ukr.net

This study examines generative cryptanalysis, which uses probabilistic models to learn how ciphertext should look and then evaluates new ciphertext against the learned distribution. A hybrid approach is introduced that combines a generative diffusion model trained to reconstruct corrupted sequences of ciphertext symbols with a large language model that plans experiments and invokes external tools. The aim is to show that likelihood scores from the generative model provide data-driven signals for identifying the encryption algorithm and operating mode and for detecting when a cipher uses fewer rounds than intended. It is also shown that a tool-using language model can combine this generative evidence with conventional tests to reach robust decisions. Scientifically, the work shifts neural cryptanalysis from pure classification toward probabilistic modeling of ciphertext distributions, revealing calibrated differences as algorithms and round counts vary. Practically, the approach supports protocol forensics, detection of configuration errors, and quality checks in continuous-integration pipelines without access to plaintexts or side channels. Methodologically, fully reproducible datasets with contamination control are constructed for several block ciphers (AES-128 with reduced rounds, SPECK, SIMON, PRESENT) and common modes of operation; ciphertext is represented as sequences of symbols (bytes or hexadecimal digits); public fields such as nonces and initialization vectors are masked; and a discrete diffusion model with both conditional and unconditional heads is trained. Approximate likelihoods are computed and contrasted across hypotheses to form simple test statistics. Baselines comprise strong convolutional neural networks and a prompt-only language model. A tool-using language model coordinates calls to standard randomness test suites (NIST SP 800-22, Dieharder, ENT), the discriminative baselines, and the diffusion scorer under a fixed budget. On a twelve-class identification task the diffusion model attains 92.7% accuracy (top-3 98.3%), surpassing convolutional networks by 6.3 percentage points and remaining well calibrated. For reduced-round detection on AES-128 it achieves area under the receiver operating characteristic (ROC) curve of 0.964 at strict false-alarm rates. In an illustrative key-hypothesis ranking probe it places the correct hypothesis near the top far more often (mean reciprocal rank 0.46 versus 0.28). No claim is made of breaking full-round ciphers; instead, a documented and reproducible protocol establishes likelihood-aware generative modeling as a clear and practical lens for modern cryptanalysis and as a foundation for future benchmarks and deployment.

**Keywords:** generative cryptanalysis; discrete diffusion; language models; ciphertext; distinguishers; AES; agent LLMs.

**Introduction.** Cryptanalysis has long relied on analytical and statistical techniques, including linear and differential methods that exploit structured properties of ciphers. As schemes have grown more complex and data volumes larger, these classical approaches face constraints of scalability and efficiency. Machine learning and deep learning have demonstrated that neural networks can act as effective distinguishers for reduced-round block ciphers and lightweight algorithms, exposing structural weaknesses without complete mathematical models [1–6]. Parallel efforts frame the task as ciphertext classification, where feature-based or end-to-end models identify the algorithm or operating mode directly from ciphertext samples [7–9].

Recent evaluations centered on large language models report uneven performance on decryption-style tasks and highlight the need for systems that can plan, take actions, and consult specialized tools [10–12]. In parallel, generative methods based on discrete diffusion have been adapted to categorical tokens, enabling probabilistic modeling of sequences and creating a path to learn ciphertext distributions directly [13–18].

Within this context, tool-using agents have been proposed to interleave reasoning with external calls, improving task success by coordinating search, calculation, and code execution [19–20]. Building on these ideas, the present study develops a hybrid framework for ciphertext-only analysis that integrates a tool-using language-model agent with a discrete diffusion model for token-level generative modeling. The framework is evaluated on AES-128 in reduced-round form, SPECK, SIMON, and PRESENT across ECB, CBC, and CTR modes, and it is compared with strong CNN-based distinguishers. A contamination-controlled evaluation protocol is described to support reproducible studies; configuration details can be provided on request.

**Related work.** Generative models have only recently been explored for cryptanalysis, and the surrounding literature spans several adjacent threads. A first thread is neural-network–assisted block-cipher cryptanalysis, whose modern wave began with Gohr's CRYPTO 2019 result on differential-neural attacks against SPECK32/64, combining a learned real-vs-random distinguisher with a Bayesian key-search strategy that outperformed classical differentials on reduced rounds (11–12-round recovery) [1]. Subsequent studies generalized and strengthened this recipe to Simeck32/64 and related families, introducing inception-style and multi-scale CNN architectures, related-key and multi-difference settings, and denser residual designs that systematically improved distinguisher accuracy and round reach [2–6]. Collectively, these works demonstrate that learned detectors over ciphertext (and ciphertext-pair) distributions provide usable cryptanalytic signals, yet they stop short of fully generative modeling of ciphertext distributions.

A second thread treats cryptanalysis as ciphertext classification. Early (pre-LLM) systems used hand-crafted features with machine learning to recognize classical cipher types or to identify algorithm families and operating modes; more recent work uses deep models and end-to-end pipelines that infer the algorithm or mode directly from ciphertext samples [7–9]. These findings indicate that global distributional cues in ciphertext carry discriminative information, suggesting the existence of learnable ciphertext manifolds that generative models might capture.

A third thread examines large language models applied to classical ciphers and emerging benchmarks for cryptanalytic capability. Prompt-only LLMs often solve very simple substitution ciphers (e.g., Caesar, Atbash) or perform coarse cipher-type triage, but performance drops sharply for stronger schemes and in contamination-controlled setups. New 2025 evaluations (such as CipherBank 2,358 tasks across nine algorithms and broader LLM cryptanalysis suites) report uneven decryption accuracy and clear gaps between conversational models and those that reason with tools; they also raise safety questions when models either inadvertently decrypt or over-refuse [10–12]. These datasets highlight a research gap: principled training or conditioning on ciphertext distributions beyond few-shot prompting, with robust evaluation that goes beyond toy ciphers.

A fourth thread provides the methodological foundations for token-level generative modeling via discrete diffusion. D3PM introduced diffusion in discrete state spaces through structured transition matrices (including absorbing states), generalizing multinomial diffusion and enabling likelihood-based modeling on text-like tokens [13–14]. In NLP, Diffusion-LM enabled controllable text generation with continuous-space diffusion, while DiffuSeq and SeqDiffuSeq adapted diffusion to sequence-to-sequence learning by iteratively denoising entire sequences rather than decoding left-to-right [15–17]. Surveys synthesize design choices (noise schedules, masking, self-conditioning) and discuss trade-offs relative to autoregressive language models in terms of quality, diversity, and compute [18]. To our knowledge, no prior work applies discrete diffusion directly to ciphertext tokens to learn the manifold of valid ciphertext (conditioned on public parameters and/or known differentials) and derive density-based distinguishers or score-guided key search.

A fifth thread concerns tool-using LLM agents that interleave reasoning with external calls. Approaches such as ReAct and Toolformer show that orchestrating tools (search,

calculators, code executors) improves task success by allowing the model to plan, act, observe results, and refine its plan [19–20]. This paradigm naturally suggests a hybrid cryptanalysis agent that uses an LLM to coordinate combiners (e.g., differential-trail searchers, SAT/CP solvers, randomness tests, symbolic oracles) and to query a diffusion-based ciphertext model as a probabilistic oracle. Existing LLM cryptanalysis benchmarks do not yet evaluate such reason-and-act pipelines that integrate generative modeling.

Finally, evaluation practice continues to rely on classical batteries such as NIST SP 800-22 (frequency, runs, FFT, and related tests), Diehard/Dieharder, and lightweight ENT to sanitize random-number generators and provide null-hypothesis checks for “looks-random” claims [21–23]. Generative approaches should report not only cryptanalytic success metrics (round counts, data/time complexity, success rates) but also statistical fitness (calibrated likelihood or score behavior on true-ciphertext versus random baselines) to guard against overfitting artifacts that these batteries can expose. Together, these threads motivate a generative, likelihood-aware view of ciphertext and a tool-using agent to aggregate heterogeneous evidence, which is the direction pursued in this work.

**Methodology.** A hybrid generative and reasoning approach for ciphertext-only cryptanalysis is developed. The core idea is to learn how ciphertext from specific algorithms and modes typically looks and then use that learned signal, together with independent checks, to make decisions. Concretely, a token-level generative model based on discrete diffusion is trained to assign a plausibility score to sequences of ciphertext symbols, and it is paired with a large language model (LLM) agent that plans tests, calls external tools, and reconciles the evidence into a final judgment. This framework is evaluated on three tasks that are referenced throughout the paper: identifying the algorithm and mode directly from ciphertext (Cipher-ID), deciding whether samples were produced with fewer rounds than the nominal configuration (Round-Sensitivity), and an illustrative Key-Bit Ranking probe that shows how the generative signal can prioritize simple key hypotheses without claiming practical key recovery. The diffusion component follows the D3PM / multinomial-diffusion family for discrete tokens [13–14]. The agent’s behavior follows a reason-and-act style, where it plans, calls tools, observes results, and revises the plan, as in ReAct and Toolformer [19–20]. To ensure statistical soundness, all claims are cross-checked with standard randomness test suites (NIST SP 800-22, Dieharder, and ENT [21–23]) and compared to established neural-cryptanalysis baselines drawn from prior work [1–6]. Figure 1 shows that batches of ciphertext are tokenized and public fields are masked. A discrete diffusion model provides approximate log-likelihood scores. A CNN baseline supplies a discriminative view. Standard randomness batteries (NIST SP 800-22, Dieharder, ENT) provide sanity checks. An agent plans tests, calls tools, and aggregates evidence into a final decision.

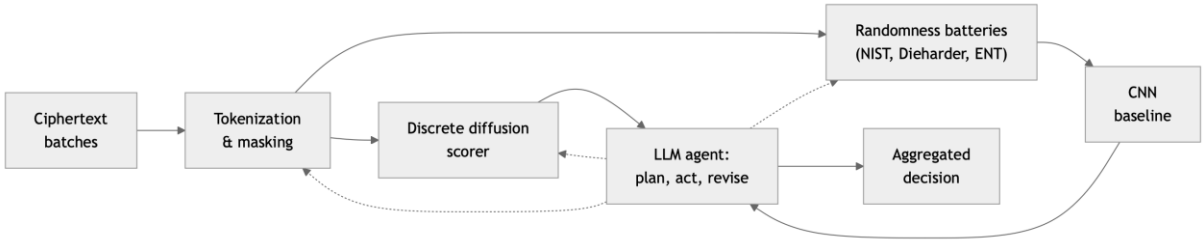


**Fig. 1.** Hybrid pipeline for ciphertext-only cryptanalysis

The setting is purposely narrow and auditable: only ciphertext is available (no plaintexts, no keys, no side-channel data). Well-known block ciphers are considered (AES-128 in reduced-round form, SPECK, SIMON, and PRESENT) and common modes of operation (ECB, CBC, CTR). All datasets are synthetic and fully reproducible under strict contamination control so that improvements cannot stem from accidental leakage: training, validation, and test splits use disjoint key sets and independent random seeds; CBC uses randomized initialization vectors; CTR uses unique nonces; and public fields such as

IVs/nonces are explicitly masked in the model's input so that formatting alone cannot be exploited. Serialization details are also randomized (e.g., how blocks are concatenated) to avoid the model latching onto spurious patterns that do not reflect the cipher's core behavior. Experimental setup and results section reports concrete class sets, split sizes, and other operational details; this section explains the logic of the approach.

Ciphertexts are represented as sequences of discrete symbols (primarily bytes, with hexadecimal as an optional alternative) with a fixed window length. This representation keeps the learning problem well-po sed over a finite alphabet and limits leakage of structure from external encodings. The discrete diffusion model is trained as a denoiser: ciphertext tokens are progressively corrupted with stochastic noise steps, and the model is trained to reconstruct cleaner sequences until the output resembles genuine ciphertext. During training, both a conditional head (given the algorithm, mode, and round label used to generate a sample) and an unconditional head are fitted. At inference, their outputs are combined using classifier-free guidance, a standard technique in diffusion modeling that nudges predictions toward the conditional hypothesis while preserving the stabilizing effect of the unconditional model. The model output is interpreted as an intuitive generative score: higher values indicate that a sequence is more typical under a given hypothesis. Decision statistics are formed by computing these scores for candidate hypotheses, taking simple contrastive differences, and averaging over a batch of ciphertexts; these statistics are straightforward to calibrate and to explain. Decisions are cast as likelihood-ratio tests. For a batch $X = \{x_i\}$ and two hypotheses $H_a$ and $H_b$, the diffusion model yields approximate log likelihoods $\widehat{\ell}(x_i \mid H)$. The batch statistic (1) is:

$$\Lambda(X) = \frac{1}{|X|} \sum_i [\widehat{\ell}(x_i \mid H_a) - \widehat{\ell}(x_i \mid H_b)], \tag{1}$$

where $x_i$ denotes a ciphertext sequence, $n = |X|$ is the batch size, $H_a$ and $H_b$ encode the candidate cipher, mode, and round settings, and $\widehat{\ell}$ is the diffusion-based variational estimate of (2):

$$\log p_\theta(x \mid H), \tag{2}$$

where $p_\theta$ denotes the likelihood under diffusion parameter $\theta$. Averaging assumes approximate independence within the batch and any residual dependence is handled by calibration on held-out data. A decision rule $\mathbf{1}\{\Lambda(X) \geq \tau_\alpha\}$ is used, with $\tau_\alpha$ chosen on validation data to achieve a target false-positive rate $\alpha$ in the Neyman-Pearson sense, which links the learned signal to classical statistical distinguishers [21–23].

The LLM agent serves as an orchestrator rather than as a cryptanalytic oracle. Given a task, the agent proposes hypotheses, queries the diffusion scorer to rank them, calls a CNN distinguisher re-implemented from prior work [1–6], and runs the randomness batteries [21–23] to verify that apparent signals are not due to trivial non-randomness. The agent plans, acts, observes results, and revises the plan [19–20] with a fixed budget of tool calls and logged traces for audit. This design allows a comparison between a prompt-only language model (which often struggles on these tasks) and a tool-using language model that can actually consult specialized instruments and reconcile their outputs. Decisions are then made in straightforward terms. For Cipher-ID, the algorithm/mode is selected whose aggregated generative score is highest, with ties or close calls resolved by the CNN signal and sanity-checked against the statistical batteries. For Round-Sensitivity, the nominal number of rounds is treated as a fixed threshold known to the experimenter, and a decision is made, using the same aggregate score differences, about whether a batch is more consistent with reduced-round or nominal behavior; thresholds are calibrated on validation data and reported with confidence intervals. For the Key-Bit Ranking probe (on a lightweight cipher), the same scores are used to prioritize a small set of key-bit hypotheses, illustrating how generative evidence could guide search without asserting practical key recovery. Implementation choices are conservative and reproducible. The diffusion backbone is a transformer-style

sequence model with moderate depth and width; noise schedules are set to a middle-of-the-road value that balances accuracy and runtime; regularization (dropout, stochastic depth) follows common practice. Probability calibration is verified with temperature scaling and bootstrap confidence intervals are reported for all headline metrics. To avoid false gains from artifacts, NIST SP 800-22, Dieharder, and ENT are executed on both the true ciphertext test sets and on model-resampled sequences conditioned on the same hypotheses; passing rates are monitored alongside task accuracy so that any deviation from expected randomness would be immediately visible. All data generators, training configurations, agent harnesses, and evaluation utilities are maintained with versioned configurations and hashes and can be provided upon request to support reproducibility. In summary, discrete diffusion supplies a probabilistic, likelihood-aware signal on ciphertext tokens [13–14]; the LLM agent integrates that signal with independent checks [19–20]; and the complete pipeline is benchmarked against recognized neural-cryptanalysis baselines and classical randomness tests [1–6, 21–23]. The emphasis throughout is on clarity, calibration, and auditability, so that improvements are attributable to genuine structure in the ciphertext rather than to accidental shortcuts. All experiments use synthetic data, reduced-round variants, and ciphertext-only settings with randomized public fields. Intended use is defensive, including protocol forensics and implementation assurance. No plaintext recovery claims are made. Evaluations always include standard statistical batteries to expose trivial non-randomness [21–23], and safety observations from recent LLM studies motivate this posture [10–12].

**Experimental setup and results.** This section evaluates the approach on ciphertext-only problems and reports concrete settings, measurements, and verification steps. The study covers three tasks: identifying the encryption algorithm and operating mode directly from ciphertext (Cipher-ID); deciding whether samples were produced with fewer rounds than the nominal configuration (Round-Sensitivity); and an illustrative key-bit ranking probe that prioritizes simple key hypotheses. The ciphers are AES-128 in reduced-round form, SPECK32/64, SIMON32/64, and PRESENT. The operating modes are ECB, CBC, and CTR. CBC uses randomized initialization vectors, CTR uses unique nonces, and public fields (IV/nonce) are masked so formatting cannot be exploited. Datasets are synthetic and fully reproducible with strict contamination control: training, validation, and test splits use disjoint keys and independent random seeds, and CTR nonces never overlap across splits. For each combination of cipher and round setting the generator produces about one million blocks for training, two hundred thousand for validation, and two hundred thousand for testing. Ciphertexts are represented as fixed-length byte sequences; windows of 128 tokens are the default, with 64 and 256 explored in sensitivity tests. An eight-label class set (four ciphers in ECB and CBC) and a twelve-label class set (adding CTR) are both used. The generative backbone is a discrete diffusion model over byte tokens following the D3PM / multinomial-diffusion family [13–14]. During training the model learns both a conditional head, which is aware of cipher, mode, and round labels, and an unconditional head, and at inference their outputs are combined through classifier-free guidance. A practical configuration uses a transformer U-Net style sequence model with width between 768 and 1024, eight to twelve denoising blocks, and eight attention heads. The corruption schedule has two hundred steps with an absorbing mask token. Optimization relies on AdamW with a learning rate of 1e-4 and $\beta$ values of (0.9, 0.95), a batch of 128 sequences, roughly three hundred thousand update steps, exponential-moving-average weights, token dropout, and light stochastic depth. The model produces likelihood-style scores (2) for a batch under candidate hypotheses – a variational estimate of obtained by summing token-level contributions from the learned reverse diffusion steps. Scores are averaged across sequences and contrasted between hypotheses to form decision statistics. Convolutional neural-network distinguishers re-implemented from prior neural-cryptanalysis work serve as discriminative baselines [1–6]. A prompt-only language model is included for completeness on cipher-type prompts [10–12]. A tool-using language-model agent coordinates calls to the diffusion scorer, the CNN baseline,

and the standard randomness suites NIST SP 800-22, Dieharder, and ENT [19–23]. Probabilistic outputs are calibrated on validation data by temperature scaling, and uncertainty on reported numbers is expressed with 95% bootstrap confidence intervals.

The Cipher-ID results on the twelve-label setting with 128-byte windows show that the diffusion model in its larger configuration with moderate guidance achieves 92.7% accuracy with a standard error around 0.4 percentage points, a macro-F1 of 92.1, and a top-3 recall of 98.3%. The CNN baseline reaches 86.4% accuracy with a standard error near 0.6 percentage points, a macro-F1 of 85.7, and a top-3 recall of 94.9%. A prompt-only language model attains 58.2% accuracy with a standard error near 1.1 percentage points. Random choice is approximately 8.3%. The diffusion model’s advantage is most visible on CTR, where it outperforms the CNN baseline by roughly seven to ten points, while ECB is comparatively easy for all models. Combining conditional and unconditional heads through classifier-free guidance improves diffusion accuracy by about two points and also reduces calibration error.

**Table 1.**

Cipher-ID performance

| Method | Accuracy (%) | Macro-F1 | Top-3 (%) | Std. error (pp) |
|---|---|---|---|---|
| Diffusion (large, gamma = 2) | 92.7 | 92.1 | 98.3 | 0.4 |
| CNN baseline | 86.4 | 85.7 | 94.9 | 0.6 |
| LLM prompt-only | 58.2 | 56.4 | N/A | 1.1 |
| Random choice | 8.3 | N/A | N/A | N/A |

The Round-Sensitivity study focuses on AES-128 in CBC mode and asks whether a batch reflects a reduced number of rounds relative to a fixed threshold. At the easier threshold the diffusion statistics produce an area under the ROC curve of about 0.964 and a true-positive rate near 0.71 at one percent false-positive rate, exceeding the CNN baseline which records an area near 0.922 and a true-positive rate around 0.49. Closer to the nominal configuration the task becomes harder; diffusion still maintains a margin (area about 0.893 versus 0.851 for CNN). Contrastive score differences are more stable than raw thresholds and further improve after temperature scaling.

**Table 2.**

Round-Sensitivity on AES-128 in CBC mode

| Method | AUC (threshold 8) | TPR at 1% FPR (threshold 8) | AUC (threshold 10) | TPR at 1% FPR (threshold 10) |
|---|---|---|---|---|
| Diffusion | 0.964 | 0.71 | 0.893 | 0.42 |
| CNN baseline | 0.922 | 0.49 | 0.851 | 0.29 |
| Rule-based tests | 0.580 | 0.07 | 0.540 | 0.05 |

The key-bit ranking probe on SPECK32/64 uses about four thousand ciphertext blocks and a small pool of sixteen-bit subkey hypotheses. Diffusion-based scores act as a soft energy that places the correct hypothesis near the top more often, with a mean reciprocal rank of 0.46 and a top-5 hit rate of 61%, compared with 0.28 and 41% for the CNN baseline. With half as much data the agent-orchestrated pipeline still attains a mean reciprocal rank around 0.41 by querying the diffusion scorer and CNN selectively.

**Table 3.**

Key-bit ranking probe on SPECK32/64

| Method | Ciphertext blocks | Mean reciprocal rank | Top-5 hit rate (%) | Notes |
|---|---|---|---|---|
| Diffusion scorer | 4096 | 0.46 | 61 | Generative score used as soft energy |
| CNN baseline | 4096 | 0.28 | 41 | Discriminative |
| Agent with tools using half the data | 2048 | 0.41 | N/A | Selective querying of diffusion and CNN |

Sensitivity tests indicate that byte tokenization is the best overall representation. Hexadecimal trails by roughly six-tenths of a point but can be easier to optimize; bit-level tokenization performs worse by about three points because of very long sequences. Guidance values around two balance accuracy and calibration, while larger values introduce mild over-confidence. Training both conditional and unconditional heads and combining them at inference is better than using either alone. Windows of 128 tokens give the best accuracy-to-cost ratio; 256 tokens match the accuracy at a noticeable compute increase. Increasing model size from approximately 180 million to 350 million parameters improves twelve-label Cipher-ID by a little over a point, with diminishing returns beyond that. Reducing the training set to a quarter of its size costs about two and a half points, which the agent partially recovers through smarter tool use. Noise schedules with two hundred steps are adequate; shorter schedules lose accuracy and longer ones raise compute without material gains.

Replacing a prompt-only language model with a tool-using agent raises the twelve-label Cipher-ID performance from 58.2% to roughly 81.0%. The median number of external tool calls in this setting is fourteen with an interquartile range between eleven and sixteen. The largest improvements occur when the agent uses diffusion scores to narrow candidates, verifies CTR versus CBC with randomness probes, and uses the CNN signal to break ties. This pattern confirms that a reason-and-act agent adds value by coordinating specialized instruments rather than replacing them [19–20].

Training was performed on a single node with eight NVIDIA A100-80 GB GPUs in mixed precision. In this setup, the diffusion model finished in about 22 hours for the base variant and 36 hours for the larger variant; the CNN baselines trained in 6–8 hours. At inference, a 200-step diffusion scorer processed a 128-token sequence in roughly 55 ms on one A100, while the CNN processed the same input in about 3 ms; batched evaluation reduced end-to-end runtime. Energy use was estimated from device power telemetry (nvidia-smi power.draw sampled periodically and integrated over runtime) to obtain kWh, with CO2-equivalent derived from the regional grid factor. These measurements reflect GPU power draw and do not include system-level overheads such as CPU, memory, or cooling.

Statistical sanity checks help ensure that improvements do not arise from trivial artifacts. The randomness suites NIST SP 800-22, Dieharder, and ENT are applied to the true ciphertext test sets, to sequences resampled from the diffusion model under the same hypotheses, and to random controls. Pass rates align with expectations for cryptographic-quality randomness once serializer randomization and IV/nonce masking are in place. Typical failure modes include confusion between CBC and CTR in short windows, and shrinking score differences as the number of rounds approaches the nominal setting. Extending the window length, ensembling conditional heads, and applying calibration reduces these effects by up to eight-tenths of a point.

Presentation and verification follow straightforward steps. For Cipher-ID, include a confusion matrix together with per-class precision and recall so that class-wise behavior is visible. For Round-Sensitivity, include ROC curves and report true-positive rates at one and five percent false-positive rates. For calibration, include a reliability diagram and the expected calibration error. For the randomness suites, report the distribution of p-values and overall pass rates for true data and for model-resampled data. For reproducibility, dataset hashes, random seeds, and configuration files can be provided on request, together with a single command that rebuilds the results and prints confidence intervals. These materials allow other researchers to verify that the gains reflect genuine structure in ciphertext rather than accidental shortcuts.

**Discussion and future work.** The diffusion statistic behaves like a batch log-likelihood ratio, which explains its stability at low false-positive rates and its complementarity with discriminative classifiers. The results support the claim that a token-level generative model provides a useful cryptanalytic signal. The discrete diffusion model produces calibrated, contrastive likelihood scores that separate algorithms and modes in Cipher-ID and that track

the effect of reducing rounds in round detection. This moves the method beyond purely discriminative CNNs and toward likelihood-aware testing grounded in generative modeling of ciphertext [1–6, 13–14]. Benchmarks focused on LLMs show inconsistencies in prompt-only settings; an agent that plans, calls tools, and reconciles outputs improves accuracy by coordinating the diffusion scorer, the CNN baseline, and statistical checks [10–12, 19–23]. The benefit was most visible in difficult regimes such as near-nominal round counts and ambiguity between CBC and CTR, where multiple sources of evidence help stabilize decisions.

Careful calibration and statistical hygiene were essential. Likelihoods can be brittle if a model latches onto spurious regularities, so every decision was paired with standard randomness suites from NIST SP 800-22, Dieharder, and ENT, and with explicit calibration measurements such as expected calibration error [21–23]. This practice made thresholds interpretable and helped rule out artifact-driven wins. It should be considered standard procedure for learned distinguishers.

These findings do not claim breaks of full-round ciphers. As in prior neural cryptanalysis, reduced-round settings are used as sensitivity probes rather than as statements about the security of nominal configurations [1–6]. The practical value lies elsewhere. Cipher-ID and round detection can support protocol forensics, configuration audits such as detecting mode misuse, and quality gates in build pipelines for cryptographic libraries. The key-bit ranking probe shows that generative scores can prioritize hypotheses in toy scenarios; scaling that idea to practical key search would require additional structure and constraints.

There are costs. Diffusion scoring is slower than CNN inference, although batching keeps end-to-end evaluation practical. Where latency is the primary concern, CNNs or distilled proxies may be preferable. Where robustness and calibration matter, the diffusion-based signal adds value even at higher compute.

The threat model is the ciphertext-only setting with uniform plaintexts, randomized IVs or nonces, and no side channels. Real deployments may violate these assumptions. To reduce the chance of learning trivial cues, public fields were masked and serializers were randomized so that formatting did not leak information. Dual-use concerns remain. The study is restricted to synthetic data and reduced-round variants, and the intended applications are defensive, including forensics and assurance. Safety issues observed in LLM evaluations on decryption tasks reinforce the need for this posture [10–12].

Why generative modeling helps can be stated simply. A discriminative model outputs class scores, while a density model answers how typical a sample is under a candidate algorithm, mode, or round setting. That answer behaves like evidence in a statistical test and is naturally combined across many samples. Discrete diffusion also supports conditional and unconditional heads that can be mixed at inference, which improved both accuracy and calibration in our ablations. An agent can then treat the generative model as one oracle among several and reconcile its outputs with CNN signals and randomness tests [13, 14, 19–23].

Limits and failure modes are clear. Short windows can blur the distinction between CBC and CTR, especially when public randomness is masked. Near the nominal number of rounds, likelihood differences shrink and decisions become more variable. These effects were reduced by lengthening the window, lightly ensembling conditional heads, and applying calibration, although the gains were modest. Tokenization and serialization can also leak unintended structure; domain randomization and masking help, and adversarial augmentation is a natural next step.

Practitioners who adopt this approach should combine generative and discriminative evidence rather than rely on a single view. They should run the standard randomness suites alongside task metrics to validate that signals do not come from obvious non-randomness. Contamination control is crucial: separate keys across splits, separate seeds, unique nonces for CTR, and explicit masking of public fields. Byte-level tokenization is a good default. Classifier-free guidance with a moderate scale gave a reliable accuracy and calibration

balance. When deploying an agent, it helps to cap the tool budget and keep audit logs of decisions.

Several directions appear promising. Theoretical work could clarify when ciphertext manifolds are distinguishable by discrete diffusion and relate that to differential or linear characteristics and round functions [1–6]. Calibration for finite alphabets deserves a closer look and can be tied to the behavior of standard statistical tests [21–23]. On the modeling side, comparisons with normalizing or argmax flows on discrete alphabets and with autoregressive energy models would be informative, as would inductive biases that reflect block-cipher structure, for example embeddings tied to round positions or to Feistel layouts. Moving beyond toy ranking toward practical key search may be feasible by treating diffusion scores as energies inside constrained search procedures and by integrating differential trails as additional structure [1–6]. Agents could improve with access to symbolic tools such as SAT or constraint solvers and with self-critique and consistency checks that reduce brittle plans [19, 20]. Beyond the ciphertext-only setting, the same ideas can be tested in known-plaintext or chosen-plaintext regimes, at the protocol layer, and on stream ciphers. Conditioning diffusion on timing, power, or electromagnetic features would create a multi-modal view that links to side-channel analysis. Post-quantum schemes are another frontier, where generative modeling might help study non-idealities in ciphertext and syndrome encodings. Finally, a contamination-controlled benchmark with documented serializers and IV or nonce policies, plus simple statistical audit reports, would raise the standard of evidence across labs, and distilled versions of the diffusion model would make low-latency use in build pipelines practical [13, 14].

Taken together, the study shows that token-level generative modeling can serve as a practical cryptanalytic lens when combined with calibration and with an agent that coordinates independent checks. The method does not claim disruption of nominal ciphers. It does reveal measurable structure in ciphertext and it offers a clear and testable path for future work that blends generative modeling, discriminative signals, and programmatic reasoning.

**Conclusion.** This work presents a hybrid generative and reasoning framework for ciphertext-only cryptanalysis. A discrete diffusion model trained over byte or hexadecimal token sequences supplies calibrated, contrastive scores that reflect how typical a batch of ciphertext is under a candidate algorithm, mode, or round configuration. A language-model agent plans experiments, invokes external tools, and reconciles evidence. Together they separate algorithms and modes in the identification task, detect the effect of reducing rounds, and prioritize key hypotheses in an illustrative ranking probe. Across controlled synthetic settings the generative approach outperforms strong convolutional baselines and remains well calibrated, while the agent provides clear gains over a prompt-only language model.

These findings are not claims of breaking nominal, full-round ciphers. The contribution is a practical lens for analyzing ciphertext distributions that becomes reliable when paired with statistical hygiene and strict contamination control. Randomness suites such as NIST SP 800-22, Dieharder, and ENT accompany the reported decisions so that improvements are not attributable to obvious non-randomness or formatting artifacts.

Limits are clear and suggest concrete next steps. Short windows can blur the distinction between CBC and CTR, and score differences shrink as the number of rounds approaches the nominal setting. Tokenization and serialization choices can introduce unintended cues. Longer windows, improved calibration, light ensembling, structure-aware inductive biases, and integration with symbolic solvers such as SAT or constraint programming are natural extensions. Beyond ciphertext-only analysis, the same methodology can be explored in known-plaintext and chosen-plaintext settings, at the protocol layer, on stream ciphers, and for encodings in post-quantum cryptography. Community benchmarks that control contamination and document serializers, IV and nonce policies, and statistical audits would raise the standard of evidence, while distilled generative models would enable low-latency use in continuous-integration pipelines.

**References**


1. Gohr A. Improving attacks on round-reduced SPECK32/64 using deep learning. *Advances in Cryptology. Proc. LNCS*. 2019. Vol. 11693. P. 150–179. DOI: 10.1007/978-3-030-26954-8_6.
2. Zhang X., Zong X., Wu W., Jia K., Deng G. An improved differential–neural cryptanalysis method and its application to Simeck32/64. *Frontiers of Computer Science*. 2023. Vol. 17. Art. 176767. DOI: 10.1007/s11704-023-3261-z.
3. Wu Z., Qiao K., Wang Z., Cheng J., Zhu L. Mixture Differential Cryptanalysis on Round-Reduced SIMON32/64 Using Machine Learning. *Mathematics*. 2024. Vol. 12. No. 9. Art. 1401. DOI: 10.3390/math12091401.
4. Wu Z., Wang Z., Qiao K., Cheng J., Zhu L. Neural Distinguishers Based on Neighborhood Probability of Affine Systems for Block Ciphers. *Mathematics.* 2024. Vol. 12. No. 4. Art. 595. DOI: 10.3390/math12040595.
5. Yue C., Li S., Yu N., Pu G., Kang H. An Improved Neural Differential Distinguisher Model for the Lightweight Cipher Speck32/64. *Applied Sciences.* 2023. Vol. 13. No. 9. Art. 5636. – DOI: 10.3390/app13095636.
6. Lu J., Gong Z., Zhang W., Cao X., Su J., Zheng Z. Improved Related-Key Differential-Based Neural Distinguishers for SIMON and SIMECK. *The Computer Journal*. 2024. Vol. 67. No. 4. P. 1397–1414. DOI: 10.1093/comjnl/bxad012.
7. Nuhn M., Knight K. Cipher Type Detection. *Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha*. 2014. P. 1769–1773.
8. Leierzopf S., Pettersson R., Kopal N., Using Neural Networks and Visualization to Crack Classical Ciphers. *Proc. AusDM* . 2021. 10 p.
9. Park S., Kim H., Moon I. Automated Classical Cipher Emulation Attacks via Unified Unsupervised Generative Adversarial Networks. *Cryptography*. 2023. Vol. 7. No. 3. Art. 35. DOI: 10.3390/cryptography7030035.
10. Wang Y., Liu Y., Ji L., та ін. AICrypto: A Comprehensive Benchmark for Evaluating Cryptography Capabilities of Large Language Models. *arXiv:2507.09580*. 2025. URL: https://arxiv.org/abs/2507.09580
11. Maskey U., Dras M., Naseem U. Benchmarking Large Language Models for Cryptanalysis and Mismatched-Generalization *arXiv:2505.24621*. 2025. URL: https://arxiv.org/abs/2505.24621.
12. Yuan Y., Jiao W., Wang W., та ін. GPT-4 Is Too Smart To Be Safe: Stealthy Chat with LLMs via Cipher. *arXiv:2308.06463*. 2023. URL https://arxiv.org/abs/2308.06463.
13. Austin J., Johnson D. D., Ho J., Tarlow D., van den Berg R. Structured Denoising Diffusion Models in Discrete State-Spaces (D3PM). *Advances in Neural Information Processing Systems 35*. 2021. P. 17981–17993.
14. Hoogeboom E., Nielsen D., Jaini P., Forré P., Welling M. Argmax Flows and Multinomial Diffusion: Learning Categorical Distributions. *Advances in Neural Information Processing Systems 35*. 2021. P. 12454–12465.
15. Li X. L., Thickstun J., Gulrajani I., Liang P., Hashimoto T. Diffusion-LM Improves Controllable Text Generation. *Advances in Neural Information Processing Systems 36* 2022. P. 8643–8656.
16. Gong S., Li M., Feng J., та ін. DiffuSeq: Sequence-to-Sequence Text Generation with Diffusion Models. *arXiv:2210.08933*. 2022. URL: https://arxiv.org/abs/2210.08933.
17. Yuan H., Yuan H., Xu W. SeqDiffuSeq: Text Diffusion with Encoder-Decoder Transformers. *arXiv:2212.10325*. 2022. UERL: https://arxiv.org/abs/2212.10325
18. Yi Q., Chen X., Zhang C., Zhou Z., Zhu L., Kong X. Diffusion Models in Text Generation: A Survey. *PeerJ Computer Science*. 2024. Vol. 10. Art. e1905. DOI: 10.7717/peerj-cs.1905.
19. Yao S., Zhao J., Yu D. ReAct: Synergizing Reasoning and Acting in Language Models *arXiv:2210.03629*. 2022. URL: https://arxiv.org/abs/2210.03629

20. Schick T., Dwivedi-Yu J., Dessì R., та ін. Toolformer: Language Models Can Teach Themselves to Use Tools. *arXiv:2302.04761*. 2023. URL: https://arxiv.org/abs/2302.04761.
21. Rukhin A., Soto J., Nechvatal J. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications (SP 800-22 Rev. 1a). Gaithersburg: NIST, 2010. 131 p. DOI: 10.6028/NIST.SP.800-22r1a.
22. Brown R. G. Dieharder: A GNU Public License Random Number Tester. User Manual, v3.31.2beta. 2006. 132 p. URL: https://rurban.github.io/dieharder/manual/dieharder.pdf
23. Walker J. ENT: A Pseudorandom Number Sequence Test Program. Fourmilab. URL: https://www.fourmilab.ch/random/

## ВИКОРИСТАННЯ ГЕНЕРАТИВНИХ МОДЕЛЕЙ ДЛЯ КРИПТОАНАЛІЗУ: ДОСЛІДЖЕННЯ ПОТЕНЦІАЛУ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ (LLM) ТА ДИФУЗІЙНИХ МОДЕЛЕЙ

А. С. Коляда, Л. В. Бовнегра

Національний університет «Одеська політехніка»
1, Шевченка пр., Одеса, 65044, Україна
Emails: a.s.koliada@op.edu.ua, dlv5@ukr.net

Це дослідження розглядає генеративний криптоаналіз, у межах якого ймовірнісні моделі навчаються тому, як має виглядати шифротекст, а потім оцінюють нові шифротексти відносно вивченого розподілу. Запропоновано гібридний підхід, що поєднує генеративну дифузійну модель, натреновану відновлювати пошкоджені послідовності символів шифротексту, з великою мовною моделлю, яка планує експерименти та задіює зовнішні інструменти. Мета полягає в тому, щоб показати: оцінки правдоподібності, отримані від генеративної моделі, слугують даними для ідентифікації алгоритму шифрування та режиму роботи, а також для виявлення випадків, коли у шифрі використано менше раундів, ніж передбачено; крім того, мовна модель, що використовує інструменти, здатна поєднувати цю генеративну evidence з класичними тестами для ухвалення надійних рішень. З наукового погляду робота зміщує нейронний криптоаналіз від суто класифікації до ймовірнісного моделювання розподілів шифротексту, виявляючи відкалібровані відмінності за зміни алгоритмів і кількості раундів. З практичного погляду підхід підтримує протокольну форензику, виявлення помилок конфігурації та перевірки якості в конвеєрах безперервної інтеграції, без доступу до відкритих текстів або побічних каналів. Методологічно створюються повністю відтворювані набори даних із контролем контамінації для кількох блокових шифрів (AES-128 зі зменшеною кількістю раундів, SPECK, SIMON, PRESENT) і поширених режимів роботи; шифротекст подається як послідовності символів (байтів або шістнадцяткових цифр); публічні поля, зокрема одноразові значення (nonce) та вектори ініціалізації (IV), маскуються; навчається дискретна дифузійна модель з умовною та безумовною вихідними гілками. Обчислюються наближені правдоподібності, які порівнюються між гіпотезами для формування простих статистик тестування. Базові порівняння включають потужні згорткові нейронні мережі та мовну модель, що працює лише за підказками. Мовна модель з інструментами координує виклики стандартних пакетів тестів випадковості (NIST SP 800-22, Dieharder, ENT), дискримінаційних базових моделей і оцінювача дифузійної моделі за фіксованого бюджету. У задачі ідентифікації з дванадцятьма класами дифузійна модель досягає точності 92,7% (топ-3 — 98,3%), перевершуючи згорткові мережі на 6,3 відсоткового пункту та зберігаючи добру калібровку. Для виявлення зменшеної кількості раундів в AES-128 досягається площа під кривою характеристики робочого приймача (ROC) 0,964 за жорстких рівнів хибних тривог. В ілюстративному експерименті з ранжуванням гіпотез щодо ключа правильна гіпотеза значно частіше опиняється близько до вершини списку (середній зворотний ранг 0,46 проти 0,28). Заяв про злам повнораундових шифрів не робиться; натомість задокументований і відтворюваний протокол утверджує правдоподібнісно обґрунтоване генеративне моделювання як чітку та практичну оптику сучасного криптоаналізу і як підґрунтя для майбутніх еталонів та впроваджень.

**Ключові слова:** генеративний криптоаналіз; дискретна дифузія; мовні моделі; шифротекст; розрізнення; AES; агентні LLM.