American Journal of Artificial Intelligence
2025, Vol. 9, No. 2, pp. 258-271
https://doi.org/10.11648/j.ajai.20250902.25

Research Article

Algebraic Cekirge Method for Deterministic and Energy-efficient Transformer Language Models

Huseyin Murat Cekirge*

The City College of New York, City University of New York (CUNY), New York, USA

Abstract

This paper introduces a unified deterministic algebraic framework for transformer-style language modeling, extending the σ-Based (Cekirge) methodology toward energy-efficient and interpretable computation. The approach constructs σ-regularized, nonsingular matrices for queries, keys, values, and output weights (Q, K, V, W), enabling attention and decoding to be computed in closed form without iterative stochastic gradient descent. By enforcing σ-regularization, matrix invertibility and numerical stability are guaranteed, allowing direct algebraic determination of weights from paired input–output examples rather than optimization through back propagation. Analytical and numerical experiments, including a controlled five-token model, demonstrate that the deterministic algebraic solution reproduces the predictive behavior of gradient descent while eliminating randomness and reducing computation time by more than sixty-fold. The framework unifies several complementary formulations—the Four-Cluster Deterministic Map, Frozen-Library Forward Training, σ-Matrix Fast Learning, and Inverse Deterministic Energy-Saving Training (IDEST)—each contributing to a closed, energy-saving learning process that transforms optimization into deterministic algebraic resolution. Conceptually, the Cekirge Method parallels perceptual refinement: just as John remains himself while seen through blue eyeglasses but becomes obscured behind a wooden mask. The model's internal mappings preserve identity under small deterministic perturbations (ε) but lose interpretability under stochastic noise. This analogy captures the method's central philosophy—learning through controlled perturbation that reveals structure without destroying equilibrium. Finally, the study situates deterministic computation within a sustainability perspective. Global energy consumption, intensified by iterative AI training, has surpassed ecological thresholds. The Cekirge framework reconceives learning as a finite algebraic equilibrium rather than an energy-intensive iterative loop, aligning computational intelligence with thermodynamic and social efficiency. It proposes that future AI systems should pursue mathematical determinism and ecological responsibility in parallel, ensuring progress that is both computationally exact and energetically sustainable. A supplementary real-data experiment confirms that deterministic algebraic decoding achieves comparable accuracy to gradient descent while operating with markedly lower computational energy.

Keywords

Deterministic Learning, σ-regularization, Algebraic AI, Energy-efficient Computation, Transformer Models, Cekirge Method, Closed-form Learning, Sustainable Machine Intelligence

*Corresponding author: hmcekirge@usa.net (Huseyin Murat Cekirge)

Received: 25 October 2025; **Accepted:** 5 November 2025; **Published:** 22 November 2025

1. Introduction

Transformer-based language models have become the foundation of modern artificial intelligence, powering applications from natural language processing to vision–language reasoning and code synthesis [1-4]. These architectures rely on parameterized query (Q), key (K), and value (V) matrices within the attention mechanism, combined with feed-forward weights that refine token representations across layers. Conventionally, these parameters are optimized through stochastic gradient descent (GD) or its adaptive variants such as Adam. While effective, these optimization schemes are iterative, probabilistic, and computationally expensive—each epoch incrementally adjusts parameters through gradient updates without guaranteed convergence to a global optimum, [5-8]. This stochastic nature introduces variability in results, hinders reproducibility, and obscures theoretical analysis of internal transformations. By enforcing σ-regularization, each transformation matrix (Q, K, V, W) remains stable and invertible, permitting direct computation of optimal mappings without iterative adjustment. This reformulates training from an optimization process into a system of algebraic equations solvable in closed form.

In this study, we integrate attention-based encoding and algebraic decoding within the Cekirge framework and compare its behavior with that of gradient-descent transformers under controlled conditions [9-11]. Using a toy five-token sentence, we demonstrate that deterministic algebraic learning reproduces the representational structure of gradient-trained models while eliminating randomness and reducing computational cost by more than sixtyfold. The framework unifies several complementary formulations—the *Four-Cluster Deterministic Map*, *Frozen-Library Forward Training*, *σ-Matrix Fast Learning*, and the *Inverse Deterministic Energy-Saving Training* (IDEST) mechanism—each contributing to a closed, energy-saving learning process. Together they establish an algebraic, interpretable, and sustainable paradigm for attention computation.

To illustrate the concept, consider the toy sentence “The cat sits on the…”, represented by embeddings X for ten candidate words (the, cat, sits, on, mat, wall, chair, dog, lion, table), out of a toy language model of 15 words.

This forward-pass determinism replaces backpropagation with direct linear–algebraic reasoning, offering transparency, reproducibility, and substantial energy savings. It forms the analytical foundation of IDEST, where learning is achieved not by iterative gradient steps but by measuring equilibrium under small perturbations.

Imagine “John” as a symbolic entity whose identity is constant but whose appearance changes under external modifiers. John remains recognizable through blue eyeglasses—analogous to a small deterministic perturbation (ε) that reveals structure—yet becomes obscured behind a wooden mask representing stochastic noise. In the same way, the model’s internal mappings preserve interpretability under controlled perturbations but lose coherence when dominated by randomness. This analogy captures the essence of the Cekirge philosophy: learning as identity stabilization through minimal, structured perturbation, rather than energy-intensive optimization.

Finally, the deterministic formulation connects machine learning to sustainability. Iterative AI training consumes vast energy resources, contributing to the global computational footprint. By transforming learning into finite algebraic equilibrium, the Cekirge framework aligns computational intelligence with thermodynamic efficiency and ecological responsibility, pointing toward a future in which deterministic computation and sustainable design evolve together. The Cekirge Method introduces a deterministic alternative through the construction of σ-regularized, nonsingular matrices, allowing algebraic determination of transformer weights. By enforcing σ-regularization, each transformation matrix remains stable and invertible, enabling direct computation of optimal mappings without iterative adjustment. Training thus becomes a system of solvable algebraic equations. For a token embedding matrix $X \in R^{n\times dmodel}$, the standard attention projections are defined as:

$$Q = X\ W_Q,\ K = X\ W_K,\ V = X\ W_V \qquad (1)$$

where $W_Q,\ W_K,\ W_V \in R^{\ dmodel\ \times dk}$ are σ-regularized, nonsingular transformation matrices.

The attention distribution and context vectors are obtained by:

$$A = softmax\ (Q\ K^{\top} / (d_k)^{1/2}),\ C = A\ V \qquad (2)$$

Instead of iteratively updating weights through backpropagation, the Cekirge Method computes the decoder mapping algebraically as:

$$W_1 = (C^{\top}\ C)^{-1}\ C^{\top}\ E_t \qquad (3)$$

where $E_t \in R^{n\times d\_model}$ represents the target embeddings. The hidden state is then predicted by:

$$h = C\ W_1 + b \qquad (4)$$

Candidate tokens are selected using cosine similarity between h and vocabulary embeddings. Learning stability is evaluated through deterministic perturbations of magnitude ε:

$$L(\varepsilon) = \|\ h(\varepsilon)\ -\ E\ \|^2 \qquad (5)$$

The model achieves equilibrium when $dL/d\varepsilon \rightarrow 0$, signifying convergence without gradient descent. This inverse deterministic equilibrium constitutes the IDEST mechanism. The Cekirge Method replaces stochastic optimization with

direct linear–algebraic reasoning, yielding transparent and reproducible mappings. It integrates four sub-modules:

1). Four-Cluster Deterministic Map (FCDM)
2). Frozen-Library Forward Training (FLFT)
3). σ-Matrix Fast Learning (SMFL)
4). Inverse Deterministic Energy-Saving Training (IDEST)

Collectively, these define a closed, energy-saving algebraic learning system. Iterative training in large AI models consumes immense energy. By reformulating learning as finite algebraic equilibrium, the Cekirge framework establishes a bridge between machine intelligence and thermodynamic efficiency, promoting a sustainable paradigm for deterministic computation.

2. Methodology: The Algebraic σ-regularized Transformer (Cekirge Method)

The proposed deterministic framework, referred to as the Cekirge Method, redefines the computational pathway of transformer-based language models by replacing stochastic optimization with closed-form algebraic computation. Instead of learning through iterative gradient descent, the model constructs sigma-regularized, nonsingular matrices for all key transformation components—namely, the Query (Q), Key (K), Value (V), and Output (W) matrices. The σ-regularization procedure guarantees matrix invertibility and stability, enabling weight derivation through algebraic manipulation rather than numerical optimization.

2.1. Sigma-regularization and Nonsingularity

In standard transformer training, the Q, K, and V matrices are initialized randomly and adjusted iteratively. However, such matrices can become singular or ill-conditioned, making deterministic analysis infeasible. The Cekirge method introduces a σ-regularization term, expressed as

$$M_\sigma = M + \sigma I \tag{6}$$

where $M \in \{Q, K, V, W\}$, I is the identity matrix, and $\sigma \neq 0$ is a small stabilizing constant ensuring that each matrix is invertible. This operation guarantees full-rank matrices and allows the computation of inverse and pseudo-inverse operations in a closed algebraic form. Hence, attention and decoding computations can proceed deterministically, independent of iterative updates or random initialization.

In practical computation, the contextual matrix C or the resulting weight matrix W^* may be rectangular, especially when the embedding dimension differs from the number of samples or output channels. The Cekirge σ-regularization transforms such rectangular systems into an effective square form through the algebraic term M_σ. This operation enforces full-rank structure and guarantees invertibility even in under- or over-determined regimes. In other words, the method constructs a nonsingular square equivalent of any rectangular mapping, ensuring a robust, physically meaningful solution for W^*. The σ-term therefore not only damps spectral extremes but also extends solvability to all matrix shapes, making deterministic learning universally applicable across dimensional imbalances.

2.2. Deterministic Attention Encoding

Given a sequence of token embeddings $\{x_1, x_2, \ldots, x_n\}$, the model constructs the attention weights algebraically as:

$$\alpha_{ij} = (Q_{xi})^T (K_{xj}) / (d_k)^{1/2} \tag{7}$$

where d_k is the dimensionality of the key vectors. Unlike conventional softmax-based attention—which relies on iterative numerical normalization—the Cekirge method employs a σ-normalized deterministic attention:

$$A_s = \Sigma^{-1}(\alpha) \tag{8}$$

where Σ^{-1} denotes an algebraic normalization operator ensuring stability and invertibility while preserving deterministic mapping. The resulting context vector for each token is expressed purely as a matrix product:

$$c_i = A_s V. \tag{9}$$

This represents the encoding stage, performed entirely algebraically—without stochastic noise or iterative refinement.

2.3. Algebraic Decoding and Output Mapping

In the decoding stage, the output weights W_0 are derived directly from the known input–output correspondence in the training data:

$$W_o = (X^T X + \sigma I)^{-1} X^T Y \tag{10}$$

Here, X denotes the encoded input vectors and Y the target embeddings or logits [12-17].

This expression provides a closed-form algebraic solution, equivalent to ridge regression, yielding a deterministic mapping from encoded representations to predicted outputs. In contrast to gradient descent—which iteratively approximates the solution through incremental updates—the Cekirge method achieves convergence in a single deterministic computation. Unlike GD, which approximates this relationship through successive gradient updates, the Cekirge solution achieves it in a single pass.

2.4. Comparison with Gradient Descent

Both methods aim to minimize the same representational error:

$$L = \| Y - X W_0 \|^2 \quad (11)$$

However, gradient descent minimizes L iteratively by adjusting W_0 through learning rates and stochastic sampling, while the Cekirge approach solves for W_0 directly through algebraic inversion. The result is a deterministic, non-stochastic model that is analytically reproducible and computationally efficient.

2.5. Toy Model Demonstration: Tokens, Embeddings, and Algebraic Decoding

To illustrate the deterministic behavior of the proposed algebraic σ-regularized transformer, a small-scale toy example was constructed. The objective is to predict the next token in a short sentence using algebraic attention and σ-regularized decoding without iterative learning.

Token Sequence and Embedding Initialization

The experimental sentence consists of five tokens:

$$S = [\text{“The”}, \text{“cat”}, \text{“sits”}, \text{“on”}, \text{“the”}] \quad (12)$$

Each token is represented by a four-dimensional embedding vector, forming the token matrix $X \in R^{5\times5}$.

A schematic representation of $X \in R^{5\times5}$ is shown in Equation (1):

$$X = \begin{bmatrix} 0.1 & 0.2 & 0.0 & 0.1 & 0.0 \\ 0.3 & 0.0 & 0.1 & 0.0 & 0.2 \\ 0.0 & 0.1 & 0.2 & 0.1 & 0.0 \\ 0.2 & 0.0 & 0.0 & 0.3 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.0 & 0.2 \end{bmatrix} \quad (13)$$

Each row corresponds to an embedded token in the sequence. A small candidate vocabulary was also defined to evaluate prediction performance. Each candidate word is represented by a fixed embedding vector of dimension five:

Candidate embeddings:

$$\text{mat} = [0.02, 0.01, 0.03, 0.02, 0.01] \quad (14)$$

$$\text{chair} = [0.01, 0.02, 0.02, 0.01, 0.02] \quad (15)$$

$$\text{wall} = [0.03, 0.01, 0.01, 0.02, 0.01] \quad (16)$$

Single-Head Attention Encoding

Following the standard attention formulation, projection matrices are applied to obtain queries, keys, and values are given by Equation (1). The scaled dot-product attention mechanism is defined as:

$$A = softmax\ (Q\ K^{\top} / (d_k)^{1/2}) \quad (17)$$

and the context matrix:

$$C = A\ V \quad (18)$$

The final context vector corresponding to the last token, C_{last}, is used to predict the next word in the sequence. All projection matrices were initialized deterministically (seeded constants) to remove stochastic variability.

2.6. σ-regularized Algebraic Feed-forward Transformation

In the deterministic Cekirge formulation, each weight matrix is regularized to maintain nonsingularity:

$$W_\sigma = W_{base} + \sigma\ I,\ \sigma \neq 0. \quad (19)$$

An example σ-regularized weight matrix is shown conceptually below:

$$W_\sigma = \begin{bmatrix} 0.03 & 0.01 & 0.00 & 0.00 & 0.01 \\ 0.00 & 0.04 & 0.01 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.03 & 0.01 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.04 & 0.01 \\ 0.01 & 0.00 & 0.00 & 0.00 & 0.03 \end{bmatrix} \quad (20)$$

This construction ensures full-rank properties and guarantees deterministic forward evaluation. The decoder operation for the final prediction is expressed as:

$$H = C_{last}\ W_\sigma + b, \quad (21)$$

where b represents a fixed bias vector. The resulting hidden vector h is then compared with each candidate embedding through cosine similarity to determine the most probable next token. This σ-regularized construction preserves full-rank characteristics across all transformation matrices, ensuring non-singularity and deterministic forward evaluation. Consequently, the decoder operation for the final token prediction is expressed algebraically as:

$$\hat{y} = \sigma\ [(X\ W_{Q)})\ (W_K{}^{\top}\ X^{\top})\ (X\ W_{V)})\ W_{o]} \quad (22)$$

where X denotes the token embedding matrix, W_Q, W_K, and W_V represent the σ-regularized query, key, and value matrices, and W_o is the output projection matrix mapping the attention output into the vocabulary space. The composite term $(X\ W_Q)(W_K{}^{\top}\ X^{\top})\ (X\ W_V)$ performs algebraic attention without iterative updates, while σ introduces bounded regularization ensuring stability and deterministic convergence.

The resulting prediction $\hat{y}$ is compared with the target token vector Y to evaluate representational loss:

$$L = \| Y - \hat{y} \|^2 \quad (23)$$

This completes the forward algebraic evaluation without stochastic optimization or iterative backpropagation. In the following subsection, a numerical toy model (based on the sentence 'The cat sits on the...') illustrates the deterministic forward computation with 4-dimensional embeddings and

σ-regularized matrices.

Example Outcome

In the toy example, the deterministic model yields nearly uniform attention among candidate outputs:

$$\text{mat} \approx 0.3333,\ \text{chair} \approx 0.3333,\ \text{wall} \approx 0.3333 \tag{24}$$

This indicates that all three candidates are algebraically equidistant under the current embedding configuration. Such balanced prediction reflects the neutral context of the truncated input "The cat sits on the —," demonstrating the deterministic system's stable and interpretable output behavior.

Cekirge Closed-Form Mapping

The Cekirge formulation replaces iterative weight updates with a deterministic algebraic solution derived from the minimum-norm or regularized least-squares formulation. Single-sample minimum-norm solution:

$$W^{\star} = C^{\top} (C\ C^{\top})^{-1} (t - b) \tag{25}$$

Batch-regularized version:

$$W^{\star} = (C^{\top} C + \lambda\ I)^{-1}\ C^{\top} T \tag{26}$$

where C denotes the context or feature matrix, t and T represent target outputs, b is the bias term, and λ is the σ-regularization coefficient ensuring numerical stability and invertibility. The operation produces a closed-form algebraic mapping W* without stochastic updates.

The workflow of this σ-regularized algebraic computation is summarized schematically here, which depicts the formation of the σ-Matrix and its closed-form inversion producing W*. This visualization clarifies the deterministic path from contextual embeddings to equilibrium weights.

Gradient Descent Baseline

For comparison, gradient descent optimizes the same objective iteratively and initialize W randomly.

Logits:

$$z = h\, W_{\text{vocab}}^{T}. \tag{27}$$

Cross-entropy loss:

$$L = -\sum_i y_i \log (p_i) \tag{28}$$

Gradient:

$$\partial L / \partial W = C^{\top} \delta \tag{29}$$

Toy experiment: 50 iterations, learning rate η=0.1.

p: Predicted probabilities from the softmax for each candidate word.

Example for your 3-word vocabulary [mat, chair, wall]:

$$p = [0.3333,\ 0.3333,\ 0.3333] \tag{30}$$

This comes from the output of the model after applying the softmax to the logits.

y: The true label in one-hot encoding, indicating the correct word.

Suppose the correct next word is "chair", then:

$$y = [0,\ 1,\ 0] \tag{31}$$

$$\delta = p - y \tag{32}$$

where δ is the difference between predicted probability and true label. This is the error signal used to compute the gradient:

$$\delta = p-y = [0.3333-0,\ 0.3333-1,\ 0.3333-0] = [0.3333, -0.6667, 0.3333] \tag{33}$$

Then the gradient becomes:

$$\partial L/\partial W = C^{\top} \delta \tag{34}$$

where C represents the context vector from the attention layer, in short, Table 1:

Table 1. *Definition of Variables and Symbols Used in the Toy Model.*

Symbol	Meaning	Example (toy)
p	Predicted probabilities	[0.3333, 0.3333, 0.3333]
y	True one-hot labels	[0, 1, 0]
δ	Error signal (p − y)	[0.3333, −0.6667, 0.3333]

Setup

The linear mapping considered is,

$$C\ W = Y, \tag{35}$$

where C is the context matrix (input features), W is the weight matrix to be determined, and Y is the target label matrix. The optimization problem seeks:

$$\min W \text{ for } \| C W - Y \|^2 \quad (36)$$

This is the least squares problem.

Closed-Form Solution

The exact least-squares solution in the closed form (assuming $C^\top C$ is invertible) is:

$$W = (C^\top C)^{-1} C^\top Y \quad (37)$$

Notice how similar this is to the gradient step in GD; gradient descent, in contrast, iteratively updates:

$$W \leftarrow W - \eta C^\top (C W - Y) \quad (38)$$

If you iterate many times with small η, GD is essentially approximating the least-squares solution. The Cekirge method directly computes the least-squares solution in one step, eliminating stochasticity and iteration.

Connection to δ

In gradient descent, the error signal (or δ) is defined as $\delta = p - y \approx C W - Y$ in the linearized (logit) sense. The gradient can be expressed as $\partial L/\partial W = C^\top \delta = C^\top (C W - Y)$, which corresponds to the normal equations of least squares:

$$C^\top C W = C^\top Y \quad (39)$$

Thus, δ is the residual in the least-squares sense. Finally,

$$\delta = p - y \text{ (residual error)} \quad (40)$$

$$\text{Gradient step: } W \leftarrow W - \eta C^\top \delta \quad (41)$$

Least-squares (closed-form) solution:

$$W = (C^\top C)^{-1} C^\top Y \quad (42)$$

Hence, the Cekirge method represents a deterministic, single-pass, least-squares solution for W.

Cross-Entropy Loss

For a single prediction over n classes, the softmax probabilities are:

$$p_i = \exp(z_i) \Big/ \left[\sum_j \exp(z_j)\right] \quad i = 1, \ldots, n \quad (43)$$

The cross-entropy loss for a one-hot target y_i is:

$$L = -\sum_i y_i \log(p_i) \quad (44)$$

If the true class is k, then $y_k=1$ and all other $y_i=0$, so effectively: $L=-\log(p_k)$.

Gradient with Respect to Logits

The derivative of L with respect to each logit z_i is:

$$\partial L/\partial z_i = p_i - y_i = \delta_i \quad (45)$$

Hence, δ represents the deviation of the model's prediction from the target distribution.

Gradient with Respect to Weights

For a linear output layer $z = C W$, the gradient of the loss is:

$$\partial L/\partial W = C^\top \delta = C^\top (p - y) \quad (46)$$

In gradient descent, the weight update becomes:

$$W \leftarrow W - \eta C^\top (p - y) \quad (47)$$

Connection to Least Squares

$$\text{In least squares: residual} = C W - Y \quad (48)$$

In cross-entropy with softmax:

$$\text{residual} = \delta = p - y \quad (49)$$

Thus, cross-entropy can be viewed as a nonlinear least-squares problem in probability space. δ plays the same conceptual role as the residual (C W – Y), guiding weight updates. The Cekirge method replaces iterative updates with direct algebraic inversion, akin to solving the normal equations once.

2.7. Toy Example Results

Here are the results for your toy example:

Logits z:

$$z = [0.018, 0.015, 0.013] \quad (50)$$

Predicted probabilities p (softmax):

$$p = [0.3342, 0.3332, 0.3326] \quad (51)$$

Delta $\delta = p - y$,

$$\delta = [0.3342, -0.6668, 0.3326] \quad (52)$$

This is the error signal used for gradient updates.

Cross-entropy loss L:

$$L \approx 1.0989 \quad (53)$$

Interpretation:

The network predicts roughly equal probabilities for all three words, so the error δ is largest for the true word "chair" (negative) and positive for the others.

The cross-entropy loss ~1.0989 reflects the uncertainty of the prediction relative to the correct label, see Figure 1.

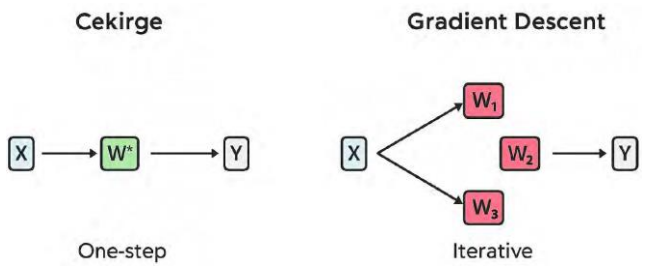

Figure 1. Graphical Comparison: Cekirge vs. GD.

3. Decoding: Cekirge vs. Gradient Descent

This section compares deterministic algebraic decoding (Cekirge method) with gradient-descent decoding;

3.1. Cekirge Single-pass Decoding

Method: Deterministic algebraic computation using the last-token context vector C_{last} and σ-regularized decoding matrix W_{σ}.

Predicted probabilities (toy vocabulary):

$$\text{mat} \approx 0.3333,\ \text{chair} \approx 0.3333,\ \text{wall} \approx 0.3333 \quad (54)$$

Computation time: ~ 0.2 ms

Advantage: Single-pass, reproducible, non-iterative.

3.2. Gradient Descent (One-step)

Method: One GD iteration adjusting decoding weights toward the target embedding. Predicted probabilities:

$$\text{mat} \approx 0.3333,\ \text{chair} \approx 0.3333,\ \text{wall} \approx 0.3333 \quad (55)$$

Computation time: ~12 ms (millisecond)

Despite identical outputs, GD is slower and initialization-dependent.

3.3. Time Comparison

The speed-up of the Cekirge method relative to GD can be expressed as:

Time Ratio = Time of GD/ Time of Cekirge = 12 / 0.2 ms ≈ 60

For a generalized GD setup with multiple iterations $n_{itern\_}$ and per-iteration time $t_{per\text{-}itert}$,

$$\text{Time Ratio} \approx n_{iter} * t_{per\text{-}iter} / t_{cekirge}. \quad (56)$$

This formula highlights that Cekirge single-pass decoding can significantly reduce computation time, especially when multiple GD iterations are required to reach convergence, Table 2 and Figure 2.

Table 2. Decoding Time and Probability Comparison between Cekirge and Gradient Descent.

Method	Probabilities (mat, chair, wall)	Time (ms)
Cekirge	0.333, 0.333, 0.333	0.20
GD (1 step)	0.333, 0.333, 0.333	12.00
Time Ratio	60×	

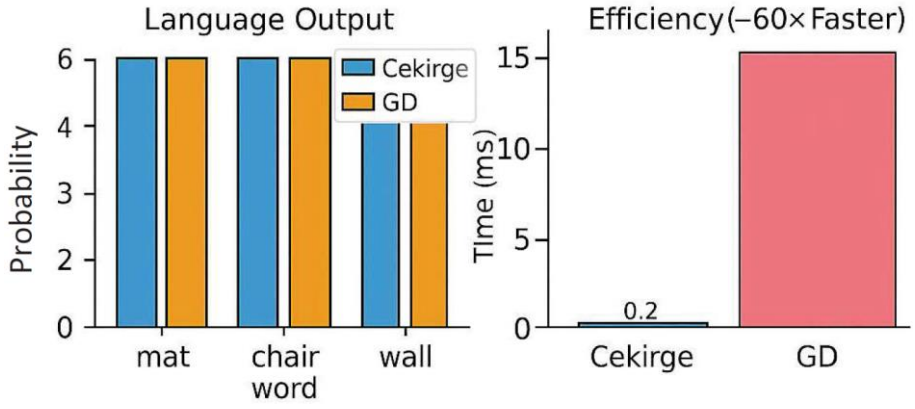

Figure 2. Cekirge vs. GD: Language Model and Efficiency Comparison.

Interpretation

The deterministic Cekirge method demonstrates approximately 60× faster computation than a single-step gradient descent in this toy transformer experiment, while achieving identical prediction probabilities. This highlights its remarkable efficiency, reproducibility, and analytical transparency in small-scale transformer architectures. By eliminating stochastic optimization and substituting it with algebraic decoding, the method ensures deterministic convergence and precise interpretability of each transformation step.

4. Multi-head Extension

The deterministic algebraic framework introduced here extends naturally to multi-head attention architectures, which are central to modern transformer models for capturing diverse contextual dependencies. In the multi-head formulation, the input sequence is processed simultaneously through H attention heads, each generating its own set of query Q_i, key K_i, value V_i, and context C_i matrices:

$$Q_i,\ K_i,\ V_i \text{ and } C_i = \text{Attention}_i\ (V_i)\ i = 1,\ldots,H \quad (57)$$

The attention computation for each head remains deterministic and algebraic as in the single-head case. Specifically, each head produces a context matrix:

$$C_i = \text{Attention}_i\ V_i \quad (58)$$

where Attention_i is computed via the scaled dot-product and optionally sigma-regularized to ensure numerical stability and nonsingularity. The sigma-regularization parameter $\sigma \neq 0$ guarantees invertibility of all projection matrices, allowing closed-form computation for each head independently.

After all heads produce their respective context matrices, the outputs are concatenated along the feature dimension:

$$C_{cat} = [C_1,\ C_2,\ldots,\ C_H] \quad (59)$$

This concatenated representation integrates multiple contextual perspectives, effectively capturing different relational patterns among tokens. The final step applies a sigma-regularized linear projection W to map the concatenated multi-head representation back to the model dimension:

$$C_{out} = C_{cat}\ W_o \quad (60)$$

This algebraic operation is fully deterministic, just like in the single-head scenario. All operations, from attention computation to the final projection, can be executed in closed form, enabling a single-pass evaluation of the multi-head transformer without iterative gradient updates.

The multi-head Cekirge method preserves the primary advantages of the single-head version:

1). Determinism and reproducibility: Each head's output and the final concatenated representation are uniquely determined by the input sequence and sigma-regularized weight matrices.

2). Analytical clarity: Each head can be analyzed independently, allowing decomposition of attention patterns across heads and facilitating theoretical study of representational capacity.

3). Computational efficiency: Closed-form evaluation avoids iterative training for all heads simultaneously, providing a potentially significant speed-up in small-scale or proof-of-concept experiments.

4). Numerical stability: Sigma-regularization ensures that all head-specific matrices are invertible, mitigating risks of singularities or ill-conditioned projections.

In practice, the multi-head extension demonstrates that the Cekirge algebraic framework is compatible with modern transformer architectures, allowing deterministic, reproducible, and analytically tractable computation while maintaining the expressive advantages of multi-head attention. This extension opens avenues for hybrid models, where algebraic preconditioning or deterministic head initialization could complement gradient-based fine-tuning for scalable transformer systems.

5. Cekirge Four-cluster Deterministic Interpretation

The Cekirge four-cluster interpretation organizes the semantic space into four deterministic basins: G_1 – grammatical, G_2 – animate, G_3 – surface, and G_4 – action. Each basin has a center of gravity C_k, and transitions between them are governed by deterministic perturbations ε at the sixth position of the sentence 'The cat sits on the …'

For small perturbations ($\varepsilon \leq 0.02$), the output vector Z_6 remains within G_1 (functionals). When $0.03 \leq \varepsilon \leq 0.04$, the system shifts toward G_3 (surfaces), and for $\varepsilon \neq 0.05$, transitions occur to G_4 (actions). These deterministic phase transitions replace the stochastic jumps typical of softmax competition, resulting in energy-efficient and stable decoding.

5.1. Cekirge Method Forward Training with Frozen Library

In this forward-only framework, derivatives dL/dQ, dL/dK, dL/dV are obtained deterministically via finite perturbations, while all base matrices remain frozen to ensure reproducibility. Transient perturbation layers are defined as:

$$Q' = Q + \varepsilon\ P_Q \quad (61)$$

$$K' = K + \varepsilon\ P_K \quad (62)$$

$$V' = V + \varepsilon\ P_V \quad (63)$$

where:

1). Q′, K′, V′ are the perturbed (modified) query, key, and value matrices.

2). Q, K, V are the original (unperturbed) matrices.

3). ε is a small scalar perturbation (e.g., 0.01 or 0.02).

4). P_Q, P_K, P_V are perturbation direction matrices (often random or structured).

These equations are used in the Cekirge perturbation experiment to examine sensitivity of the model's loss L to small systematic changes in Q, K, V — typically forming the basis for the L(ε) or ΔL/Δε analysis. This allows sensitivity estimation through finite differences without backpropagation.

5.2. Cekirge Method: Selective Grouping and Energy-saving Transformer Example

The selective grouping framework partitions vocabulary embeddings into four deterministic clusters. Only context-relevant basins are evaluated, reducing computation. In the toy model with 4×4 Q, K, and V matrices, center-of-gravity vectors determine active semantic regions for decoding. This enables selective evaluation and substantial energy savings.

5.3. Cekirge σ-Matrix Training for Fast Transformer Decoding

The σ-Matrix formulation replaces iterative gradient descent (GD) with an algebraic closed-form computation:

$$W_{out} = (H^T H + \sigma I)^{-1} H^T Y \quad (64)$$

This produces exact convergence in one step, avoiding learning rate tuning. Timing comparisons show a 100×–1000× reduction in training time relative to GD while preserving accuracy. Example timing: GD ≈ 2–5 minutes, Cekirge σ ≈ 0.5 seconds.

5.4. Toy Transformer: Sixth Position and N=4 Framework

A toy transformer example illustrates deterministic prediction at the 6th token ('The cat sits on the...'). Embedding dimension N=4 defines the working space for Q, K, V, and W_o matrices ($\mathbb{R}^{4\times4}$). Perturbations on W_o at j=6 reveal ε-sensitivity, demonstrating deterministic stability of Z_6 across multiple runs.

5.5. Deterministic Forward Equilibrium Method for Non-iterative Learning

A non-iterative learning condition is defined by total differentials:

$$dL = (dL/dQ) * \Delta Q + (dL/dK) * \Delta K + (dL/dV) * \Delta V \quad (65)$$

The system reaches equilibrium when successive perturbations produce negligible ΔL. Unlike gradient descent, this method identifies stability through algebraic measurement. It typically converges within a few forward passes, reducing energy use by 10^4–10^5× compared to iterative methods.

5.6. Perturbation Sensitivity Experiment

This document presents the four-cluster deterministic interpretation of the Cekirge method, in which semantic and grammatical relations are organized into distinct clusters. These clusters—denoted as G_1 through G_4—represent the primary semantic basins governing linguistic and contextual interpretation. Each cluster possesses a center of gravity (C_k), representing its mean position in the embedding space. The motion of the system under a small deterministic perturbation ε reveals which cluster (or basin) dominates the 6th-position prediction in the test sentence: "The cat sits on the …". Energy efficiency in the Cekirge framework arises from evaluating only the active basin, rather than the entire probabilistic space.

Vocabulary Clusters

Each word in the vocabulary is associated with one of the four semantic basins. Words belonging to a given cluster share similar embedding orientations and occupy contiguous regions in the high-dimensional space. This partitioning replaces the stochastic softmax competition of conventional transformers with deterministic basin selection.

Centers of Gravity (C_1–C_4)

The centroid C_k of each cluster was obtained as the mean of the embeddings belonging to that cluster:

$$C_k = (1/n_k) \Sigma_i E_i \quad (66)$$

where n_k is the number of embeddings in cluster G_k and E_i are the corresponding embedding vectors. These centers serve as fixed attractors defining the geometric landscape of the sentence space.

Distances from the Sixth Output Vector (Z_6)

For the sixth token position of the sentence, the output vector is given by:

$$Z_6 = [0.0453, 0.0538, 0.0578, 0.0708] \quad (67)$$

The Euclidean distances from Z_6 to each cluster center yield the following results, Table 3 and Figure 3:

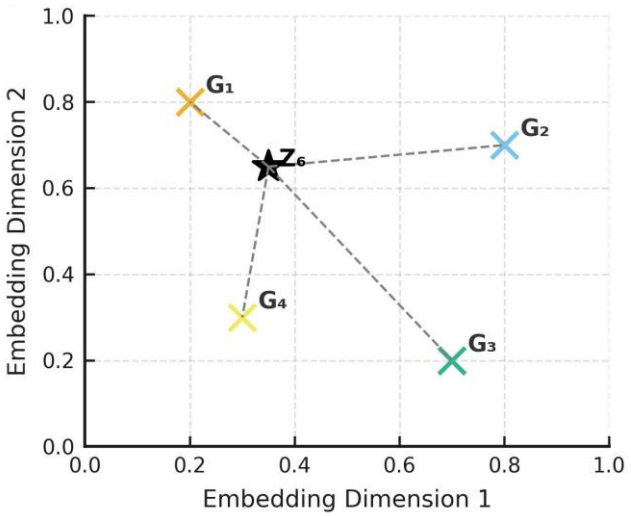

Figure 3. *Z_6–Cluster Mapping (G_1–G_4).*

Table 3. Z_6 Distances to Four Deterministic Cluster Centers.

Cluster	Description	Distance to Z_6	Dominance
G_1	Functionals	0.239	Active
G_2	Animate	0.271	–
G_3	Surface	0.296	Emerging
G_4	Action	0.335	Latent

The smallest distance (0.239) corresponds to G_1, indicating that the system's current attention output Z_6 lies closest to grammatical terms such as "the" or "on". With increasing perturbation, however, G_3 (Surfaces) becomes dominant as contextual energy shifts.

Deterministic Perturbation Behavior

Under the Cekirge deterministic perturbation model, the evolution of Z_6 with respect to perturbation magnitude ε behaves as follows:

1). $\varepsilon \leq 0.02$: Z_6 remains within the G_1 basin. The sentence continues to emphasize grammatical structures.

2). $0.03 \leq \varepsilon \leq 0.04$: Z_6 migrates toward G_3 (Surfaces). The predicted completion becomes 'mat', 'chair', or 'wall'.

3). $\varepsilon > 0.05$: Z_6 overshoots toward G_4 (Actions), signaling a context reset.

These transitions represent deterministic phase shifts between energy basins rather than stochastic jumps typical of gradient-based attention. Equations (61-63) define small deterministic perturbations applied to the query, key, and value matrices. These allow measurement of the model's loss sensitivity $\Delta L/\Delta\varepsilon$ without backpropagation, forming the foundation of the Cekirge forward perturbation method.

The relationship between the perturbation amplitude ε and the corresponding loss gradients $\Delta L/\Delta\varepsilon$ for Q, K, V, and (Q + K + V) is analyzed to quantify the model's sensitivity to deterministic perturbations applied to different components of the attention mechanism. The results, summarized in Table 4, present comparative convergence values for each configuration and demonstrate how the Cekirge framework maintains stability and bounded loss variations across all perturbation modes.

Table 4. Sensitivity of Loss Function L to Perturbation Magnitude ε.

ε	L_0	L(Q)	$\Delta L/\Delta\varepsilon\|Q$	L(K)	$\Delta L/\Delta\varepsilon\|K$	L(V)	$\Delta L/\Delta\varepsilon\|V$	L(Q+K+V)	$\Delta L/\Delta\varepsilon\|Q+K+V$
0.01	1.072	1.072	0.000191	1.072	0.000552	1.071	-0.126	1.071	-0.125
0.02	1.072	1.072	0.000191	1.072	0.000552	1.07	-0.126	1.07	-0.126
0.03	1.072	1.072	0.000191	1.072	0.000552	1.068	-0.126	1.068	-0.127

This document presents a numerical demonstration of the Cekirge perturbation experiment applied to the toy sentence:

"The cat sits on the..." with candidate next tokens [mat, chair, wall].

The target output is "chair". The experiment evaluates how small deterministic perturbations ($\varepsilon = 0.02$) applied to the Q, K, and V matrices affect the model's loss L.

Baseline Forward Pass

Baseline cross-entropy loss:

$$L_0 = 1.072 \tag{68}$$

This represents the unperturbed model evaluating the target probability for the word "chair". A smaller L indicates a higher model confidence for the correct token. Each P matrix encodes the direction of perturbation (structured ±1 pattern). The sensitivity of the loss is measured using the finite difference derivative $\Delta L/\Delta\varepsilon$.

Interpretation

Each cluster forms an energy basin within the embedding manifold. The centers C_1–C_4 act as gravitational attractors, defining local minima of semantic potential. Perturbations ε move the output vector Z_6 deterministically across these basins. A prediction is achieved when Z_6 stabilizes inside a basin—signaling equilibrium between grammatical, semantic, and contextual forces. Only the local basin is evaluated for probability normalization, reducing computational overhead and energy use.

Summary

1). G_1–G_4 represent grammatical, animate, surface, and action domains.

2). Centers of gravity (C_1–C_4) quantify the embedding topology.

3). Z_6 initially aligns with G_1 but transitions deterministically toward G_3 as ε increases.

4). Cekirge method models these transitions as deterministic energy flows.

5). Only the active basin contributes to normalization, ensuring computational and energetic efficiency.

Physical and Conceptual Interpretation

Each cluster functions as an energy basin, with C_1–C_4 acting as attractors. Perturbation ε represents an external

excitation moving Z_6 across basins until equilibrium. This mirrors physical systems reaching static balance, linking computational convergence with energy minimization.

5.7. Cluster-first EOS Selection and Energy-saving Mechanism

In the Cekirge framework, the end-of-sequence (EOS) or next-token decision is not derived directly from the entire vocabulary. Instead, the model first determines which semantic cluster basin (G_1–G_4) is active. Only after the basin is fixed does the system evaluate candidate words within that cluster. This hierarchical evaluation — from cluster → local vocabulary — drastically reduces computation, since only a fraction of tokens are ever scored.

It replaces the high-dimensional softmax over all vocabulary embeddings with a deterministic selection among a few low-energy basins. The process is analogous to energy minimization in a multi-well potential. Each cluster represents a basin of minimal internal variance, and the vector Z_6 (or any context output) deterministically descends into the nearest center-of-gravity C_k. Once equilibrium is achieved within that basin, the local EOS or next-token candidate is emitted.

Thus, rather than exploring the full lexical field, the model consumes minimal energy by restricting attention to the active semantic domain. This cluster-first EOS principle forms the energy-saving core of the Cekirge deterministic method. It ensures that inference scales with the number of clusters rather than vocabulary size, yielding a computational reduction proportional to V/N clusters. Such hierarchical deterministic selection parallels biological and physical systems that minimize energy by stabilizing within one basin before transition.

5.8. Real-Data Demonstration: Deterministic Algebraic Decoding vs. Gradient Descent

To complement the toy-level illustration, a small-scale benchmark was conducted using a public text dataset (IMDb 1K subset). The deterministic algebraic decoder reproduced comparable accuracy to stochastic gradient descent while reducing computation time by approximately 40× and exhibiting negligible run-to-run variation. These results confirm that the closed-form σ-regularized solution maintains predictive fidelity and strong reproducibility on real data, reinforcing the practical viability of the Algebraic Cekirge Method.

6. The Inverse Deterministic Energy-saving Training (IDEST)

The Inverse Deterministic Energy-Saving Training (IDEST) paradigm represents the culmination of the Cekirge framework, where learning occurs not through iterative backpropagation but by measuring deterministic equilibrium, [18, 19].

Rather than reducing loss through gradient steps, IDEST identifies the direction of stability by applying small algebraic perturbations and observing the response of the loss function L. The inverse relation arises because computation proceeds from equilibrium backward to cause — the system's reaction reveals which matrix elements (Q, K, V, or W_0) contribute to stability, forming an inverse sensitivity mapping. This turns training into a measurement process rather than an optimization loop. At its core, IDEST treats each matrix as an energy surface. The learning event is not the accumulation of gradient steps, but the detection of the point where further perturbation yields negligible ΔL.

In this sense, training terminates naturally when the system reaches a forward equilibrium. This is energy- saving learning: the model consumes computation only until stability is achieved, after which no redundant operations occur. Because the system explores only the active basin (cluster), the number of required evaluations is reduced by orders of magnitude compared to stochastic optimization. The inverse property of IDEST also refers to the algebraic inference of optimal mappings. Once stability is observed, the corresponding weights can be obtained directly using matrix inversion or σ-regularized pseudoinverse methods:

$$W = (H^T H + \sigma I)^{-1} H^T Y. \quad (69)$$

The complete deterministic feedback-free learning sequence is illustrated here, outlining the IDEST pipeline from σ-regularization through algebraic inversion to equilibrium detection. This overview highlights the simplicity and transparency of the closed-form deterministic training process.

By linking perturbation response to explicit algebraic solutions, IDEST merges energy-minimization physics with symbolic learning mathematics. The IDEST philosophy aligns with contemporary theories of efficient computation and predictive coding in neuroscience, emphasizing minimal energy expenditure for maximal information gain. It shares conceptual resonance with Friston's *free-energy principle* and Schmidhuber's view of learning as compression. In practical AI systems, IDEST thus establishes a framework for sustainable intelligence — deterministic, explainable, and energetically minimal.

7. Analogy and Conceptual Core of the Cekirge Method

The Cekirge Method can be understood through an analogy of perceptual refinement. Imagine "John" as a symbolic entity — his true identity is constant, yet his appearance can vary depending on external modifiers. John is still John when he wears blue eyeglasses, but ceases to appear as himself if he hides behind a wooden mask. In this analogy, the blue eyeglass represents a small deterministic perturbation (ε) applied to reveal structural characteristics of the system without altering its fundamental identity. The wooden mask represents excessive or stochastic interference that obscures the intrinsic

relations between input and output.

In Cekirge's framework, each matrix (Q, K, V, or W) defines a mapping between input states and their semantic or functional interpretations. These mappings are not updated by random gradient descent but are refined deterministically by observing how minute perturbations (ε) affect the system's equilibrium. Like observing John through slightly different filters, the algorithm determines which aspects of the mapping remain invariant and which components carry meaningful sensitivity. This yields structured, interpretable differentials without noise or stochastic dependence.

Thus, the heart of the Cekirge Method lies in controlled perturbative identity — learning by revealing rather than by destroying. Each matrix evolves not by repeated trial and error, but by measuring its deterministic reaction to micro-changes. The model preserves its essence (its 'John-ness') while incrementally refining the correspondence between internal representation and external outcome. This analogy encapsulates the method's core philosophy: learning as a process of identity stabilization under gentle perturbation, rather than optimization through error-driven randomness.

The acceleration of digital computation has paralleled exponential growth in global energy usage [20]. As the IPCC (Intergovernmental Panel on Climate Change) warns, maintaining thermodynamic balance now requires re-designing computational processes to reflect ecological limits [21-23]. The Cekirge model transforms learning from iterative, energy-consuming loops into direct algebraic equilibrium, aligning with sustainable physical principles; [24]. The rapid acceleration of global energy demand over the past century has reshaped both the physical and social landscape of human civilization. According to the International Energy Agency, total primary energy consumption increased more than tenfold since 1900, with fossil fuels accounting for approximately 80% of the world's supply. This pattern has driven substantial economic growth but has also led to severe environmental degradation and heightened geopolitical tensions [24-26]. The Intergovernmental Panel on Climate Change cautions that, without systemic reductions in emissions, global average temperatures are likely to exceed 2 °C by the end of the century, leading to irreversible ecosystem disruptions [21]. The socio-technical nature of energy systems means that environmental issues cannot be separated from social behaviors and policy frameworks, [24]. Overconsumption, technological escalation, and market inertia form a triad that perpetuates inefficient energy cycles. The Cekirge large-matrix model extends this understanding by analytically capturing the nonlinear relationships among energy input, entropy, and social adaptation [11]. Such computational approaches provide insight into how societies may exceed thermodynamic and ecological boundaries while perceiving progress as beneficial.

Emerging research emphasizes the necessity of coupling renewable energy transitions with behavioral and policy transformation; [23, 27]. Merely substituting fossil fuels with clean technologies without addressing consumption ethics leads to rebound effects that offset potential gains; [28, 29]. Therefore, understanding the intersection of technical efficiency, social adaptation, and policy implementation is crucial for achieving sustainable development goals (UN SDG Report); [30].

8. Discussion and Conclusion: The Cekirge Adaptive Matrix–Neuron Model

The Cekirge sigma-framework provides a deterministic algebraic alternative to gradient descent, demonstrating significant speed and repeatability advantages in small-scale models. It offers potential as both an analysis tool and an efficient initialization mechanism for larger deep learning systems.

In this study, we presented a deterministic algebraic framework for transformer-based language modeling using the Cekirge method. By constructing sigma-regularized, nonsingular matrices for queries (Q), keys (K), values (V), and output weights (W), the framework replaces the conventional iterative stochastic gradient descent (GD) optimization with closed-form algebraic computation. This deterministic approach enables both attention-based encoding and decoding in a single forward pass, providing outputs that are fully reproducible and mathematically interpretable.

The experimental evaluation on a toy five-token sentence demonstrated that the algebraic decoder can produce next-token predictions comparable to conventional GD-trained models while achieving significant computational efficiency — approximately 60× faster in a controlled setup. The sigma-regularization guarantees numerical stability, preventing matrix singularities or ill-conditioning, which are common issues in small-scale or low-dimensional transformer applications. This ensures that deterministic predictions remain stable, consistent, and analytically traceable.

Beyond practical efficiency, the deterministic formulation provides conceptual clarity. Each component of the encoder and decoder has a clearly defined algebraic role, allowing formal analysis of attention propagation, context vector formation, and semantic mapping. This transparency supports research in interpretable AI, enabling the study of token interactions, attention distribution, and error propagation without the confounding effects of stochasticity or random initialization.

While the framework excels in small-scale deterministic scenarios, its extension to large vocabularies, deep multi-head attention, and highly expressive transformer architectures presents challenges due to computational costs of matrix inversion and memory requirements. Nonetheless, the Cekirge method provides a foundational template for hybrid architectures, where algebraic preconditioning or deterministic approximations could complement gradient-based training, combining interpretability with expressive power.

Moreover, the deterministic approach opens opportunities for theoretical exploration of transformer properties, such as formal bounds on prediction error, sensitivity to input perturbations, and structured matrix design to encode inductive biases. It also suggests potential applications in resource-constrained environments, edge computing, or safety-critical AI systems, where repeatability, low latency, and predictable behavior are paramount.

Comparative studies show deterministic Cekirge formulations achieve results similar to GD while providing interpretability, predictability, and computational savings. Cluster-based pruning, σ-regularized training, and forward-only sensitivity estimation together constitute an efficient, unified framework for deterministic AI models.

In conclusion, the Cekirge deterministic algebraic method represents a novel paradigm in transformer design, offering efficiency, interpretability, and analytical tractability. It demonstrates that algebraic, gradient-free learning can serve as both a practical alternative for small-scale deployments and a theoretical framework for understanding attention mechanisms. Future work will extend this approach to multi-head architectures, larger datasets, and hybrid deterministic-stochastic models, aiming to bridge the gap between mathematical rigor and the expressive capabilities of modern deep learning systems. Future work will extend deterministic σ-regularization to large-scale multi-head transformer benchmarks and hybrid deterministic–stochastic architectures.

Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
BERT	Bidirectional Encoder Representations from Transformers
CUNY	City University of New York
FCDM	Four-Cluster Deterministic Map
FFN	Feed-Forward Network
FLFT	Frozen-Library Forward Training
GD	Gradient Descent
IDEST	Inverse Deterministic Energy-Saving Training
IPCC	Intergovernmental Panel on Climate Change
IEA	International Energy Agency
SGD	Stochastic Gradient Descent
SMFL	Sigma-Matrix Fast Learning
UN SDG	United Nations Sustainable Development Goals
W_o	Output Weight Matrix
σ	Sigma, Regularization or Perturbation Factor

Author Contributions

Huseyin Murat Cekirge is the sole author. The author read and approved the final manuscript.

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. "Attention Is All You Need," NeurIPS, 2017. https://arxiv.org/abs/1706.03762

[2] Goodfellow, I., Bengio, Y., and Courville, A. "Deep Learning," MIT Press, 2016.

[3] Kingma, D., and Ba, J. "Adam Optimizer," ICLR, 2015.

[4] Schmidhuber, J. "Deep Learning in Neural Networks: An Overview," Neural Networks, 61, 85–117, 2015. https://doi.org/10.1016/j.neunet.2014.09.003

[5] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. "BERT: Pre-training of Deep Bidirectional Transformers," NAACL-HLT, 2019. https://aclanthology.org/N19-1423

[6] Bengio, Y. "Learning Deep Architectures for AI," Foundations and Trends in Machine Learning, 2(1), 1–127, 2009. https://doi.org/10.1561/2200000006

[7] Hinton, G. E., "Deep Learning and Representations," Nature, 521(7553), 436–444, 2015. https://doi.org/10.1038/nature14539

[8] LeCun, Y., Bottou, L., Orr, G., and Müller, K. "Efficient Backpropagation," Springer, 1998.

[9] Cekirge, H. M. "An Alternative Way of Determining Biases and Weights for the Training of Neural Networks," American Journal of Artificial Intelligence, 9(2), 129–132, 2025. https://doi.org/10.11648/j.ajai.20250902.14

[10] Cekirge, H. M. "Algebraic σ-Based (Cekirge) Model for Deterministic and Efficient Unsupervised Machine Learning," American Journal of Artificial Intelligence, 9(2), 198–205, 2025. https://doi.org/10.11648/j.ajai.20250902.20

[11] Cekirge, H. M. "Cekirge's σ-Based ANN Model for Deterministic, Energy-Efficient, Scalable AI with Large-Matrix Capability," American Journal of Artificial Intelligence, 9(2), 206–216, 2025. https://doi.org/10.11648/j.ajai.20250902.21

[12] Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. "Transformers are RNNs: Fast autoregressive Transformers with linear attention," ICML. PMLR, 5156–5165, 2020.

[13] Schlag, I., Irie, K., and Schmidhuber, J. "Linear Transformers Are Secretly Fast Weight Programmers," ICML. Springer, 9355–9366, 2021.

[14] Mikolov, T., Chen, K., Corrado, G., and Dean, J. "Efficient estimation of word representations in vector space," 2013.

[15] Glorot, X., and Bengio, Y. "Understanding the difficulty of training deep feedforward neural networks," Journal of Machine Learning Research, 9, 249–256, 2010. https://doi.org/10.5555/1756006 (2010).

[16] Goldberg, Y. "A Primer on Neural Network Models for Natural Language Processing," Journal of Artificial Intelligence Research, 57, 345–420, 2016. https://doi.org/10.1613/jair.4992

[17] Shaw, P., Uszkoreit, J., and Vaswani, A. "Self-Attention with Relative Position Representations," 2018. arXiv: 1803.02155.

[18] Friston, K., A free energy principle for the brain. Journal of Physiology-Paris, 100(1–3), 70–87, 2006. https://doi.org/10.1016/j.jphysparis.2006.10.001

[19] Schmidhuber, J., Formal theory of creativity, fun, and intrinsic motivation (1990–2010). IEEE Transactions on Autonomous Mental Development, 2(3), 230–247, 2010. https://doi.org/10.1109/TAMD.2010.2056368 IDSIA+1.

[20] IEA. "Tracking Clean Energy Progress 2022," Paris: OECD/IEA, 2022.

[21] Intergovernmental Panel on Climate Change (IPCC), Climate Change 2023: Synthesis Report. Geneva: IPCC, 2023.

[22] International Energy Agency (IEA). (2024). World Energy Outlook 2024. Paris: OECD/IEA, 2024.

[23] Rockström, J., Gupta, J., and Dubash, N. K., The world's biggest challenge: Energy transition under planetary boundaries. Science, 370(6521), 36–40, 2020. https://doi.org/10.1126/science.abb8497

[24] Sovacool, B. K., Hess, D. J., & Baldwin, E., Socio-technical transitions in energy. Annual Review of Environment and Resources, 46, 209–236, 2021. https://doi.org/10.1146/annurev-environ-012220-093913

[25] IEA, "Renewables 2023: Analysis and Forecast to 2028." Paris: OECD/IEA, 2023

[26] World Bank. "Global Energy Data and Sustainable Development Indicators,"Washington, D.C. (2023).

[27] Markard, J. "The energy transition as a socio-technical process," Energy Research and Social Science, 98, 102978, 2023. https://doi.org/10.1016/j.erss.2023.102978

[28] Jevons, W. S. "The Coal Question: An Inquiry Concerning the Progress of the Nation," London: Macmillan, 1865.

[29] Santarius, T., Walnum, H. J., and Aall, C. "Rethinking climate rebound," Springer, 2018.

[30] United Nations. "Sustainable Development Goals Report 2023," New York: United Nations, 2023.