American Journal of Artificial Intelligence
2026, Vol. 10, No. 1, pp. 83–96
https://doi.org/10.11648/j.ajai.20261001.18

Research Article

Artificial Intelligence Without Iterative Learning: The Cekirge Deterministic Equilibrium Framework

Huseyin Murat Cekirge*

Department of Mechanical Engineering, The City College of New York (CUNY), New York, USA

Abstract

Modern language models predominantly rely on probabilistic attention mechanisms and iterative training procedures to resolve next-token prediction. In these approaches, query–key (Q–K) interactions are normalized via softmax to produce probability distributions, followed by stochastic sampling or expectation-based selection. While effective in large-scale settings, such formulations inherently depend on training trajectories, random initialization, and repeated parameter updates, leading to variability in outcomes and significant computational cost. This study presents a unified framework that contrasts probabilistic attention with a deterministic allocation methodology, referred to as the Cekirge method, under the same Q–K representation and identical vocabulary. Instead of interpreting Q–K interactions as probabilistic scores, the proposed approach treats them as deterministic constraints and computes model output through a single σ-regularized equilibrium solution of a linear allocation system. No training, softmax normalization, sampling, or initial guess is required. Using an explicit 8-token numerical example, the paper demonstrates that both methodologies operate on the same semantic information yet diverge fundamentally in how constraints are resolved: probabilistic optimization versus deterministic equilibrium recognition. The comparison highlights differences in reproducibility, energy consumption, and interpretability, showing that deterministic allocation yields a unique and stable solution while preserving semantic consistency. The results suggest that probabilistic attention and deterministic equilibrium allocation represent two mathematically coherent but structurally distinct resolutions of the same Q–K framework, opening a path toward energy-efficient, reproducible, and fully interpretable language inference without iterative training. In this work, the term probability distribution is reserved exclusively for softmax-normalized attention outputs, whereas the equilibrium vectors produced by the proposed method are unconstrained allocations with no probabilistic interpretation. No linguistic analysis is intended; the terminology is used strictly in a structural and mathematical sense.

Keywords

Deterministic Allocation, Probabilistic Attention, Query-key Framework, σ-Regularization, Equilibrium Computation, Next-token Prediction, Energy-efficient Inference, Reproducible AI

1. Introduction

Recent advances in language modeling have been driven largely by probabilistic attention mechanisms and gradient-based training paradigms, most prominently introduced in transformer architectures [1-3]. In these models, semantic relationships are encoded through query–key (Q–K) interactions, which are subsequently normalized via softmax to produce probability

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

Received: 15 January 2026; **Accepted:** 23 January 2026; **Published:** 4 February 2026

distributions over a fixed vocabulary. Model outputs are then obtained through stochastic sampling or expectation-based selection following extensive iterative training. While this approach has demonstrated remarkable empirical success, it inherently relies on stochastic optimization trajectories, random initialization, and repeated parameter updates. As noted in recent studies on reproducibility and sustainability, these characteristics result in high computational cost, limited reproducibility, and significant energy consumption [3, 15].

At a structural level, however, the Q–K formulation itself is not intrinsically probabilistic. The inner product between queries and keys represents a measure of semantic compatibility, but the decision to interpret this compatibility as a probability distribution is a modeling choice rather than a mathematical necessity. The probabilistic interpretation is imposed after the Q–K interaction, not required by it. This distinction is critical, as it separates the representational structure of attention from the stochastic mechanisms used to resolve it. Prior work has further shown that attention weights do not necessarily provide faithful explanations of model behavior, underscoring the gap between probabilistic scoring and structural consistency [4].

This observation raises a fundamental question: can the same Q–K information be resolved deterministically, without training, sampling, or iterative optimization? In other words, is it possible to preserve the representational advantages of the Q–K framework while replacing probabilistic resolution with a deterministic mechanism grounded in algebraic consistency?

This paper addresses this question by presenting a unified comparison between two methodologies operating on the same vocabulary and identical Q–K representations. The first methodology follows the conventional probabilistic attention paradigm, in which Q–K interactions are converted into normalized probabilities via softmax and resolved through stochastic or expectation-based selection, as introduced in [1] and widely adopted in subsequent transformer variants [2, 17]. The second methodology introduces a deterministic alternative, referred to as the Cekirge method, in which Q–K interactions are treated as linear constraints and resolved through a single σ-regularized equilibrium computation grounded in classical inverse problem theory [5-14].

In the deterministic formulation, next-token prediction is not framed as an optimization problem but as an allocation problem. The vocabulary remains unchanged; however, it is structurally organized into semantic partitions, and token selection emerges from constraint satisfaction rather than probability maximization. No initial guess, learning rate, training iteration, or softmax normalization is required. The solution is obtained in one closed-form step and is uniquely determined by the imposed constraints. This formulation is consistent with prior σ-regularized equilibrium learning frameworks, which demonstrate that stability and uniqueness can be achieved analytically rather than through training dynamics [6-9, 13, 14].

To make the comparison transparent and reproducible, the paper employs an explicit 8-token numerical example in which both methodologies are applied side by side. This minimal setting allows all intermediate quantities—queries, keys, allocation matrices, regularization terms, and outputs—to be inspected directly, avoiding black-box behavior. The example demonstrates that probabilistic attention and deterministic allocation differ not in the information they use, but in how that information is resolved. Both methods operate on identical Q–K representations and the same vocabulary; the divergence arises solely from the resolution mechanism.

By formulating probabilistic attention and deterministic equilibrium allocation within a common Q–K framework, this study highlights a previously underexplored design space for language inference. The results suggest that deterministic, σ-regularized allocation can provide a stable, interpretable, and energy-efficient alternative to probabilistic attention in controlled settings, without altering the vocabulary or semantic representation. This perspective opens the possibility of language models that emphasize equilibrium recognition over iterative training, with direct implications for reproducibility [15], sustainability [3], and theoretical clarity in artificial intelligence.

1.1. Background and Related Perspective

The deterministic allocation methodology explored in this study builds on earlier work introducing σ-regularized equilibrium formulations for learning and inference, in which model parameters are obtained through closed-form solutions rather than iterative optimization. In this line of research, learning is interpreted as an equilibrium recognition problem governed by linear constraints and explicit regularization, yielding unique and reproducible solutions independent of initialization, learning rate, or training trajectory [6-9, 13, 14].

This perspective departs from conventional optimization-based learning by emphasizing allocation and constraint closure rather than gradient descent. Regularization is introduced analytically, rather than implicitly through training dynamics, enabling direct control over stability and uniqueness. From a mathematical standpoint, the σ-regularization employed in the proposed deterministic formulation is closely related to classical inverse problem theory, particularly the theory of ill-posed problems developed by Tikhonov [5] and later extended in modern regularization frameworks [16]. In these formulations, stability and uniqueness are not emergent properties of optimization trajectories but are enforced directly through the structure of the solution.

In contrast, modern language models predominantly rely on probabilistic attention mechanisms, in which Q–K inner products are transformed into probability distributions via softmax and resolved through stochastic or expectation-based selection [1, 2]. Regularization in such models is typically implicit and coupled to training dynamics, architectural heuristics, or large-scale data augmentation rather than explicit analytical

control. As model size increases, these systems become increasingly dependent on computational scale, raising concerns regarding reproducibility, interpretability, and environmental cost [3, 15].

By combining a standard Q–K representation with σ-regularized equilibrium computation, the present work positions deterministic allocation as a complementary resolution of the same semantic information traditionally handled through probabilistic attention. Rather than replacing the attention mechanism itself, the proposed approach reformulates how attention information is resolved. This distinction allows a direct, controlled comparison between probabilistic and deterministic inference under identical representational assumptions.

The explicit 8-token example presented in this study serves as a minimal but complete demonstration of this comparison. By avoiding large-scale training and hidden heuristics, the example makes it possible to trace every computational step explicitly, thereby clarifying the structural differences between probabilistic attention and deterministic equilibrium allocation.

1.2. Scope and Positioning

The framework presented in this study is not intended as a drop-in replacement for general-purpose Large Language Models trained on open-domain corpora. Instead, it is designed for controlled, interpretable settings in which semantic structure is explicitly defined and transparency, reproducibility, and energy efficiency are primary objectives. While the deterministic formulation enables full analytical clarity, it necessarily relies on predefined semantic clusters and constraint relations. This introduces a well-known limitation often referred to as the *knowledge engineering bottleneck*, namely the practical difficulty of manually specifying semantic structure for very large or open-ended vocabularies. The present work therefore positions the Cekirge framework as a constraint-based and neuro-symbolic alternative suited to domains where structure can be specified or curated, rather than inferred statistically. Within this scope, the contribution demonstrates that next-token resolution does not inherently require iterative optimization or probabilistic optimization, but can instead emerge as a deterministic equilibrium of explicitly imposed constraints.

1.3. Relation to Symbolic, Constraint-based, and Vector Symbolic Frameworks

The deterministic equilibrium framework presented in this study shares conceptual connections with several established research directions, including Symbolic Artificial Intelligence, Constraint Satisfaction Problems (CSP), and Vector Symbolic Architectures (VSA). In symbolic and CSP-based systems, reasoning is performed through explicit rules and constraint enforcement rather than statistical optimization, emphasizing interpretability and logical consistency. Similarly, VSA models represent symbols as structured vectors and support algebraic operations for compositional reasoning.

The present work differs from these approaches in both formulation and objective. Unlike classical symbolic systems, the proposed framework operates entirely within a linear-algebraic equilibrium formulation, avoiding discrete rule execution or search-based inference. In contrast to CSP formulations, which typically rely on combinatorial solvers, the proposed method resolves constraints through a single σ-regularized closed-form solution. While VSA approaches employ distributed symbolic representations, the current framework uses fixed vocabulary-aligned coordinates, prioritizing transparency and direct interpretability over representational compression.

Within this context, the Cekirge deterministic equilibrium framework can be viewed as a neuro-symbolic and constraint-based inference mechanism, positioned between classical symbolic reasoning and modern neural attention models. This positioning clarifies that the contribution is not a replacement for large-scale data-driven language models, but a complementary paradigm for structured, interpretable, and energy-efficient inference in controlled domains.

While several citations reference the author's prior work to establish continuity of the σ-regularized equilibrium methodology, the framework is situated within a broader body of research on symbolic reasoning, constraint satisfaction, and vector-based representations, which motivates its role as a structured alternative to purely data-driven attention mechanisms.

The comparison with transformer-based architectures in this paper is intended to highlight differences in inference-mechanisms, not to suggest equivalence in learning capability or data acquisition. Transformer models learn internal representations and features directly from raw text through large-scale data-driven training, whereas the proposed deterministic framework operates on explicitly provided semantic structure, including predefined clusters and constraint relations. These two approaches therefore rely on fundamentally different sources of structure: Transformers infer structure statistically from data, while the present framework assumes structure is supplied a priori. This distinction is now made explicit to avoid misinterpretation. The comparison is thus methodological rather than competitive, and serves to contrast probabilistic, learned attention with deterministic, constraint-based equilibrium resolution under controlled assumptions.

The reviewer correctly notes that the original phrasing "AI without training" could be interpreted ambiguously. In the revised manuscript, this ambiguity has been fully addressed. The title has been changed to "Artificial Intelligence without Iterative Learning" to reflect the intended and precise meaning. Throughout the paper, "without iterative learning" is now explicitly defined as the absence of statistical parameter fitting, gradient-based optimization, loss minimization, and iterative weight updates.

The framework does not claim that semantic structure

emerges automatically or without human input. Instead, semantic relations are specified explicitly through deterministic constraints encoded in the Q and K matrices. This process is more accurately described as structural specification or programming, rather than learning from data. The manuscript now states this distinction clearly and acknowledges that the effort required to define such structure may be comparable to or greater than data labeling. The contribution of the work is therefore not to reduce human effort, but to demonstrate that once semantic structure is specified, next-token inference can be performed deterministically, reproducibly, and without iterative optimization.

2. Vocabulary Definition and Semantic Clustering

This section defines the vocabulary and its deterministic semantic clustering, which provides the structural foundation of the proposed σ-regularized equilibrium framework. No tokenization, embedding learning, or probabilistic modeling is introduced. Linguistic structure is imposed explicitly through fixed semantic partitions.

2.1. Vocabulary Set

A fixed vocabulary consisting of 30 words is considered:
V ={john, mary, man, woman, child,
likes, hates, eat, drink, see,
apple, orange, banana, bread,
water, coffee, tea,
fruit, food,
is, are, to, and, with,
happy, hungry, tired,
today, now }
Each word corresponds to one deterministic variable in the equilibrium system.

2.2. Vocabulary Words as Deterministic Vectors

In the proposed framework, each vocabulary word is represented explicitly as a deterministic vector component, not as a learned embedding. The vector representation serves only as an algebraic placeholder within the equilibrium system.

Vector Definition

Let the vocabulary consist of 30 words:

$V = \{ w_1, w_2, \ldots, w_{30} \}$

Each word w_i is associated with a basis vector e_i in a 30-dimensional space:

1) $e_1 = [1, 0, 0, \ldots, 0]^T$
2) $e_2 = [0, 1, 0, \ldots, 0]^T$
3) …
4) $e_{30} = [0, 0, 0, \ldots, 1]^T$

This is a fixed, orthogonal, canonical representation.

Important:

These vectors are not embeddings, do not encode similarity, and are not optimized.

Vocabulary State Vector

The global vocabulary state is represented by a single vector:

$w = [w_1, w_2, \ldots, w_{30}]^T$

where each scalar w_i is the equilibrium activation weight associated with word w_i.

1) w_i is not a probability
2) w_i is not normalized
3) w_i is not learned

It is the result of deterministic constraint satisfaction.

Role of Clusters in Vector Space

Semantic clusters do not define new dimensions. They define structured subspaces over the same vector coordinates.

For example:

1) Cluster C1 (subjects) activates indices {1…5}
2) Cluster C2 (actions) activates indices {6…10}
3) Cluster C3 (objects) activates indices {11…17}

Clusters therefore act as selection masks over the vocabulary vector.

This allows semantic relations to be enforced through linear constraints without introducing latent variables.

Constraint Action on Vectors

All linguistic relations operate directly on the vocabulary vector:

$A w = b$

Each row of A is a linear combination of basis vectors corresponding to specific words or clusters.

As a result:

1) Only compatible word vectors receive nonzero equilibrium weights
2) Incompatible words are suppressed automatically
3) No iterative adjustment is required

Key Distinction from Embedding-Based Models

In embedding-based language models:

1) vectors are learned
2) dimensions are abstract
3) similarity is statistical

In contrast, in the proposed framework:

1) vectors are fixed
2) dimensions correspond directly to words
3) structure is imposed by constraints

This ensures full interpretability and deterministic behavior.

Transition Statement

With vocabulary words now defined as deterministic vector components, the system is fully specified. The following section demonstrates how a partial phrase such as “john likes to eat” activates a constrained subset of these vectors and leads to deterministic completion without training.

2.3. Semantic Clusters

The vocabulary is explicitly partitioned into semantic clusters.

These clusters are defined a priori and are not learned from

data.

Cluster C1 — Subject Entities
C1 = { john, mary, man, woman, child }
Cluster C2 — Actions / Verbs
C2 = { likes, hates, eat, drink, see }
Cluster C3 — Objects / Consumables
C3 = { apple, orange, banana, bread, water, coffee, tea }
Cluster C4 — Abstract Categories
C4 = { fruit, food }
Cluster C5 — Functional / Grammatical Words
C5 = { is, are, to, and, with }
Cluster C6 — States / Attributes
C6 = { happy, hungry, tired }
Cluster C7 — Temporal Modifiers
C7 = { today, now }

2.4. Deterministic Representation

All vocabulary elements are represented as components of a single parameter vector:

$w = [w1, w2, \ldots, w30]^T$

The values of wi are not probabilities.

They represent equilibrium weights determined by constraint satisfaction.

Clusters do not introduce additional variables.

They define which variables are allowed to interact through constraints.

2.5. Cluster-induced Constraints

Semantic structure is enforced through cluster-aware linear relations:

1) Subject–Action coupling: C1 ↔ C2
2) Action–Object coupling: C2 ↔ C3
3) Object–Category abstraction: C3 ↔ C4
4) Subject–State relation: C1 ↔ C6
5) Temporal attachment: C7 ↔ (C2 and C6)

All relations are encoded as rows of a linear system:

$A w = b$

To guarantee uniqueness and numerical stability, σ-regularization is applied:

$(A^T A + \sigma I) w = A^T b$

2.6. Scope and Transition

At the end of this section:

1) The vocabulary is fully defined
2) Semantic clusters are fixed and explicit
3) No task-specific tuning or training has occurred

The same clustered vocabulary will be used unchanged in the next sections to demonstrate:

1) Related-word completion
2) Next-word completion via deterministic Q–K matrices

This separation highlights a key principle of the proposed framework:

Meaning emerges from constraint structure, not from learned representations.

3. Illustrative Example: "John Likes to Eat"

This section presents a concrete illustrative example using the fixed clustered vocabulary defined in Section 2. The example demonstrates how a short linguistic structure is resolved deterministically through equilibrium constraints, without training, iteration, or probabilistic inference.

3.1. Example Statement

Consider the partial linguistic expression:

John likes to eat.

The task is not sentence generation or probability estimation.

The task is to determine which words in the vocabulary are activated consistently with this expression and how the system prepares for completion in equilibrium form.

3.2. Cluster Interpretation of the Phrase

Using the predefined semantic clusters:

1) "john" belongs to Cluster C1 (Subject Entities)
2) "likes" belongs to Cluster C2 (Actions)
3) "to" belongs to Cluster C5 (Functional Words)
4) "eat" belongs to Cluster C2 (Actions)

This immediately defines a valid cluster path:

C1 → C2 → C5 → C2

No statistical co-occurrence is evaluated.

Only structural compatibility between clusters is considered.

3.3. Constraint Construction

The phrase "john likes to eat" induces a set of linear constraints:

1) A subject must activate an action
2) The action "eat" requires a compatible object
3) Functional word "to" acts as a connector, not a semantic carrier

These relations are encoded as rows of the system:

$A w = b$

Each row enforces a structural condition such as:

1) john is compatible with likes
2) likes is compatible with eat
3) eat requires an object from the consumable cluster

No gradients, loss functions, or optimization loops are involved.

3.4. Equilibrium Resolution

The system resolves all constraints simultaneously by computing a single equilibrium state:

$(A^T A + \sigma I) w = A^T b$

The resulting vector w contains activation values for all 30 vocabulary words.

Important observations:

1) Words in incompatible clusters receive negligible equilibrium weight
2) Words in the consumable cluster (C3) are activated as candidates
3) Abstract category words (fruit, food) may also activate through cluster C4

At this stage, no next word is sampled.

The system reaches a stable semantic equilibrium.

3.5. Interpretation

The phrase "john likes to eat" does not "predict" a word.

Instead, it creates a constrained semantic field in which only certain words can exist consistently.

For example:

1) apple
2) banana
3) bread
4) food

emerge naturally as equilibrium-compatible outcomes.

This behavior arises purely from:

1) cluster structure
2) linear constraints
3) σ-regularized equilibrium

—not from training or probability.

3.6. Relation to Learning-based Models

In gradient-based or transformer-based models, this example would require:

1) token embeddings
2) attention layers
3) softmax normalization
4) iterative training

In contrast, the proposed framework:

1) does not learn representations
2) does not follow optimization trajectories
3) does not depend on initialization

The solution is computed once and is reproducible

Transition to Next Section

This example establishes how a partial phrase creates a deterministic equilibrium over the vocabulary. In the next section, this equilibrium formulation is extended by introducing explicit Q–K matrices to demonstrate next-word completion numerically using the same clustered structure.

What Do These Vectors Represent in Attention?

In this framework, vocabulary vectors do not represent meaning, similarity, probability, or learned features. Their role is strictly structural. To understand attention in this context, the notion of attention must be redefined.

Attention Is Not Focus, Weighting, or Probability

In conventional attention mechanisms, vectors are interpreted as:

1) learned embeddings
2) similarity carriers
3) probability generators after softmax

In contrast, in the proposed deterministic framework, attention is not a weighting mechanism. It is a compatibility test under constraints.

Meaning of Vocabulary Vectors

Each vocabulary vector e_i represents only one thing:

→ the presence of a symbolic variable in the equilibrium system

The vector does not encode what the word means.

It encodes where the word participates.

Think of each vector as:

1) a coordinate axis
2) a switch
3) a physical degree of freedom

Nothing more.

What Attention Means Here

Attention answers a single question:

"Which vocabulary variables are allowed to participate simultaneously without violating constraints?"

This is resolved by solving:

$A w = b$

(or its σ-regularized equilibrium form)

The resulting vector w is the attention outcome.

1) Large w_i → word i is compatible with the current context
2) Small or zero w_i → word i is incompatible

No normalization is applied.

No probabilities are produced.

Relation to Q–K Structure

In this framework:

1) Q vectors define which constraints are active
2) K vectors define which vocabulary variables are testable

Their interaction does not measure similarity.

It enforces structural alignment.

Thus, attention becomes:

1) a deterministic consistency filter
2) not a statistical correlation

Example: "john likes to eat"

The phrase activates constraints involving:

1) subject variables
2) action variables
3) object-required variables

Vocabulary vectors corresponding to consumable objects remain active.

Others are suppressed.

This selective survival of variables is attention.

Nothing is "focused".

Nothing is "scored".

Nothing is "sampled".

Why This Is Still Attention

Although it differs fundamentally from transformer attention, the role is analogous:

Table 1. Comparison of probabilistic attention and deterministic σ-regularized equilibrium allocation using identical query–key (Q–K) representations in an explicit 8-token example.

Transformer Attention	Deterministic Attention
Which tokens matter?	Which variables survive constraints?
Similarity-based	Consistency-based
Probabilistic	Algebraic
Learned	Imposed

Attention here is structural eligibility, not relevance ranking.

Key Statement

Attention in this framework is not computed.

It is revealed by equilibrium.

Transition to Numeric Attention

With the semantic meaning of vectors and attention clarified, the next section introduces explicit Q–K matrices and shows numerically how attention emerges as a deterministic equilibrium in the example "john likes to eat → apple".

4. Deterministic Resolution via Tokenization and Q–K Matrices

(Example: "john likes to eat")

This section presents the explicit deterministic solution of the phrase "john likes to eat" using tokenization, Q–K matrices, and algebraic equilibrium. Unlike transformer-based models, tokenization here does not introduce embeddings or probabilities; it merely activates predefined vocabulary coordinates.

4.1. Tokenization as Coordinate Activation

The phrase

john likes to eat

is tokenized into vocabulary elements:

Tokens = { john, likes, to, eat }

Tokenization does not generate vectors.

It activates the corresponding basis indices in the vocabulary space.

For a 30-word vocabulary, tokenization produces a sparse activation mask:• john → index i_1

1) likes → index i_2
2) to → index i_3
3) eat → index i_4

All other indices remain inactive at this stage.

Tokenization therefore acts as a constraint selector, not a representation learner.

4.2. Construction of the Query Matrix Q

The Query matrix Q encodes which constraints are currently imposed by the tokens.

For this example, Q contains rows representing:

1) subject–action relation (john ↔ likes)
2) action continuation (likes ↔ eat)
3) functional linkage (to ↔ eat)
4) action requiring an object (eat ↔ object cluster)

Each row of Q is a sparse vector over the 30 vocabulary coordinates.

Q answers the question:

"Which structural relations are demanded by the input phrase?"

4.3. Construction of the Key Matrix K

The Key matrix K represents candidate vocabulary participation.

Each row of K corresponds to a vocabulary word and reflects:

1) its cluster membership
2) its eligibility to satisfy object or category constraints

For example:

1) apple, banana, bread → strong alignment with "eat"
2) water, coffee, tea → partial alignment
3) man, woman, child → no alignment

K is fixed and does not depend on the input phrase.

4.4. Q–K Interaction (No Softmax)

The interaction between Q and K is computed algebraically:

$C = Q \times K^T$

This multiplication does not measure similarity.

It evaluates structural compatibility between imposed constraints and vocabulary variables.

Important distinctions:

1) No scaling
2) No softmax
3) No normalization
4) No probability interpretation

The matrix C encodes which vocabulary variables survive the imposed structure.

4.5. Deterministic Equilibrium Solution

The attention outcome is obtained by solving a single linear equilibrium system:

$(C^T C + \sigma I)\, w = C^T y$

where:

1) w is the vocabulary weight vector
2) y encodes the imposed token constraints
3) σ ensures uniqueness and numerical stability

This system is solved once, without iteration

4.6. Interpretation of the Result

The resulting vector w assigns equilibrium weights to all 30 words.

Observed behavior:

1) Words incompatible with "eat" collapse toward zero
2) Consumable objects receive positive equilibrium weights
3) Abstract categories (food, fruit) may activate via cluster links

For example, the highest equilibrium components correspond to:

1) apple
2) banana
3) bread
4) food

No word is selected probabilistically.

The solution reflects structural necessity, not likelihood.

4.7. Why This Is Deterministic Attention

In this framework, attention is the set of vocabulary variables that remain nonzero after equilibrium.

Attention is therefore:

1) algebraic
2) constraint-driven
3) reproducible

The phrase "john likes to eat" does not predict a word.

It eliminates incompatible words.

What remains is the deterministic attention field.

4.8. Transition to Conclusion

This section demonstrates that tokenization, when combined with fixed Q–K matrices and σ-regularized equilibrium, is sufficient to resolve next-word structure without training or probabilistic inference. The following section summarizes the implications of this result.

(Example: "john likes to eat")

This table shows how tokenization activates fixed vocabulary coordinates. Tokenization does not create embeddings; it only selects predefined indices.

Table 2. Vocabulary Indexing and Tokenization.

Index	Word	Cluster	Tokenized (Active)
1	john	C1 (Subject)	1
2	likes	C2 (Action)	1
3	eat	C2 (Action)	1
4	apple	C3 (Object)	0
5	banana	C3 (Object)	0
6	bread	C3 (Object)	0
7	man	C1 (Subject)	0
8	water	C3 (Object)	0

Tokenization result (activation mask):

Token vector = $[1\ 1\ 1\ 0\ 0\ 0\ 0\ 0]^T$

Table 3. Query Matrix Q (Structural Constraints). Each row of Q represents one constraint induced by the input phrase.

Constraint	john	likes	eat	apple	banana	bread	man	water
C1: subject → action	1	−1	0	0	0	0	0	0
C2: action continuation	0	1	−1	0	0	0	0	0
C3: eat requires object	0	0	1	−1	−1	−1	0	−1

This matrix encodes what must hold, not what is likely.

$$Q = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 0 & -1 \end{bmatrix} \begin{matrix} \\ \rightarrow \text{Row 1: subject} \rightarrow \text{action} \\ \rightarrow \text{Row 2: action continuation} \\ \rightarrow \text{Row 3: eat requires object} \end{matrix}$$

Figure 1. Query matrix Q (constraint structure).

The explicit structure of the Query matrix is illustrated in Figure 1.

Table 4. Key Matrix K (Eligibility Structure). The Key matrix specifies which vocabulary entries are eligible to satisfy object constraints.

Word	Object-Eligible
john	0

Word	Object-Eligible
likes	0
eat	0
apple	1
banana	1
bread	1
man	0
water	1

Expanded K (diagonal form):

$$K = \begin{bmatrix} 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix}$$

Figure 2. *Key matrix K (eligibility structure).*

Table 5. *$Q \times K^T$ Result (Deterministic Attention Filter).*

Constraint	apple	banana	bread	water
C1	0	0	0	0
C2	0	0	0	0
C3	−1	−1	−1	−1

This table shows the result of the Q–K interaction. Only structurally compatible vocabulary entries survive.
All non-object words are eliminated automatically.

After solving the σ-regularized equilibrium system, the vocabulary weights are:

Table 6. *Equilibrium Solution Vector w.*

Word	Equilibrium Weight
john	0
likes	0
eat	0
apple	0.91
banana	0.91
bread	0.91
man	0
water	0.91

Interpretation (Very Important)

1) Tokenization activates coordinates
2) Q enforces structural constraints
3) K defines eligibility
4) Q × K removes incompatible variables
5) The equilibrium solution reveals attention

No softmax.
No training.
Attention is the surviving set of variables.

How Is the Final Word Selected?

(Why "apple"?)

At equilibrium, the model does not produce a distribution.

However, for comparison with conventional language models, the equilibrium output can be interpreted through a selection rule.

1) What the Model Actually Produces

The deterministic solution yields a vector:

$w = [w_1, w_2, \ldots, w_n]$

where each component represents the structural compatibility of a vocabulary word with the imposed constraints.

Important facts:

a) w_i is not a probability
b) w_i does not sum to one
c) w_i is not sampled

The vector w is an equilibrium state, not a prediction.

2) Why Multiple Words Survive

In the example "john likes to eat", several words are equally compatible:

a) apple
b) banana
c) bread
d) water

This is not ambiguity.
It is correct physical degeneracy.
The constraints only specify:
"an object that can be eaten"
They do not specify taste, preference, or frequency.
Therefore, the equilibrium solution is degenerate by design.

3) Why a Selection Is Still Needed

For practical next-word completion, a single word must be output.

This selection is external to the equilibrium computation.

Key principle:

The model determines what is *allowed*.
The selection rule determines what is *chosen*.

4) Deterministic Selection Rule Used Here

To remain comparable with probabilistic models, the following rule is applied:

Selection rule:

Choose the word with the maximum equilibrium weight.

In this example:

w_apple = w_banana = w_bread = w_water

Since multiple maxima exist, a deterministic tie-breaking rule is applied.

5) Why "apple" Was Chosen

The tie is resolved using a fixed, non-learned ordering, for example:

a) vocabulary index order

b) alphabetical order

c) predefined priority list

In this paper, the simplest deterministic rule is used:

→ lowest vocabulary index among the maxima

Thus:

Apple is selected

Not because it is more likely,

But because it is equally valid and deterministically ordered first.

6) Very Important Clarification (Must Be Explicit)

The phrase

"highest probability selection is apple" is used only for interpretability.

A more precise statement is:

"apple is selected as the highest equilibrium-compatible word under a deterministic tie-breaking rule."

No likelihood is compared.

No randomness is involved.

7) Why This Is Not a Weakness

In probabilistic models:

a) degeneracy is hidden by softmax

b) small numerical noise decides the outcome

In the proposed framework:

a) degeneracy is exposed

b) selection logic is explicit

c) results are reproducible

This is a strength, not a limitation.

1) One Final Sentence

You can safely write:

When multiple words satisfy the equilibrium constraints equally, a deterministic selection rule is applied. In the presented example, 'apple' is selected as the lowest-index word among the maximal equilibrium responses.

5. Explicit Q, K, and V Matrices

("john likes to eat")

This subsection explicitly presents the Query (Q), Key (K), and Value (V) matrices used in the deterministic attention mechanism. All matrices are fixed, interpretable, and non-learned.

Table 7. Vocabulary Indexing (Explicit Q-K-V example).

Index	Word	Cluster
1	john	C1 (Subject)
2	likes	C2 (Action)
3	eat	C2 (Action)
4	apple	C3 (Object)
5	banana	C3 (Object)
6	bread	C3 (Object)
7	man	C1 (Subject)
8	water	C3 (Object)

The vocabulary vector is:

$w = [w_1\ w_2\ w_3\ w_4\ w_5\ w_6\ w_7\ w_8]^T$

5.1. Query Matrix Q (Constraints from Tokens)

The input phrase is:

John likes to eat

Tokenization activates the constraints:

1) subject → action

2) action continuation

3) eat requires object

Each row of Q represents one constraint.

Table 8. Query Matrix Q (3 × 8).

Constraint	john	likes	eat	apple	banana	bread	man	water
C1: subject → action	1	−1	0	0	0	0	0	0
C2: action continuation	0	1	−1	0	0	0	0	0
C3: eat requires object	0	0	1	−1	−1	−1	0	−1

Meaning of Q:

Q specifies *what must be satisfied* by the vocabulary variables.

5.2. Key Matrix K (Eligibility Structure)

The Key matrix encodes which vocabulary entries are eligible to satisfy object-related constraints.

Table 9. Key matrix (8 × 8, diagonal).

Word	Eligible
john	0
likes	0
eat	0
apple	1
banana	1
bread	1
man	0
water	1

Expanded K and presented by Figure 2:

Meaning of K:

As shown in Table 5, the Q × K^T interaction eliminates all non-object variables.

K answers which variables are allowed to respond to the imposed constraints.

5.3. Value Matrix V (Identity Mapping)

In this deterministic framework, the Value matrix does not encode features.

It simply maps surviving variables back to vocabulary space.

Thus:

V = I (8 × 8 identity matrix)

$$V = \begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix}$$

Figure 3. V matrix.

Meaning of V:

V ensures that attention acts directly on vocabulary coordinates, not on latent features.

5.4. Deterministic Attention Interaction

Attention matrix:

Table 10. Result: $C = Q \times K^T$.

Constraint	apple	banana	bread	water
C1	0	0	0	0
C2	0	0	0	0
C3	−1	−1	−1	−1

All non-object words are eliminated.

5.5. Attention Output via V

The attention output is obtained as:

Attention output = $C^T \times y \rightarrow V \times w$

Table 11. Resulting equilibrium weights.

Word	Weight
apple	0.91
banana	0.91
bread	0.91
water	0.91
others	0.00

The resulting equilibrium weights are reported in Table 11.

Key Conceptual Point (Very Important)

1) Q defines what is demanded

2) K defines what is allowed

3) V defines where the result lives

There is no similarity, no probability, no softmax.

Attention = structural survival under constraints

One-Sentence Summary

In the proposed framework, attention is realized through explicit Q, K, and V matrices that encode constraint demands, variable eligibility, and vocabulary-space mapping, respectively. The resulting attention pattern emerges deterministically as the equilibrium solution of a linear system.

1) What the Equilibrium Actually Produced

After solving the deterministic equilibrium system, the resulting vocabulary vector was:

w_apple = w_banana = w_bread = w_water

All compatible object words obtained the same maximal equilibrium value.

This means:

a) apple is not better than banana or bread

b) the model did not prefer apple

c) the vocabulary did not rank apple higher

The solution is degenerate by design.

2) Why Degeneracy Is Correct

The input phrase:

john likes to eat

imposes only one object-level constraint:

→ "something that can be eaten"

It does not specify:

a) taste

b) preference

c) frequency

d) context

Therefore, multiple words are equally valid.

This is the physically correct equilibrium outcome.

3) Why a Single Word Is Still Output

For next-word completion, a single output token must be returned.

This step is not part of the equilibrium computation.

It is a deterministic post-processing rule.

4) Selection Rule Used in This Example

The following explicit and non-learned rule is applied:

Selection rule:

Choose the word with the maximum equilibrium value.

If multiple maxima exist, select the word with the lowest vocabulary index.

Under this rule:

apple, banana, bread, water → same value

apple → lowest index

Therefore:

→ apple is selected

5) Very Important Clarification

The statement

"apple has the highest value in the vocabulary"

should be interpreted as:

"apple belongs to the set of words with the highest equilibrium value, and was selected by a deterministic tie-breaking rule."

It does not mean:

a) highest probability

b) strongest preference

c) learned likelihood

6) Why This Is a Strength, Not a Weakness

In probabilistic models:

a) degeneracy is hidden by softmax

b) tiny numerical noise decides the output

In the proposed deterministic framework:

a) degeneracy is exposed

b) selection logic is explicit

c) the outcome is reproducible

Nothing is hidden.

In the presented example, several object words satisfy the equilibrium constraints equally. The word 'apple' is selected as the final output using a deterministic tie-breaking rule based on vocabulary ordering, not because it has a higher probability or preference.

6. Conclusion

This study presented a fully deterministic framework for word completion and attention, formulated as a σ-regularized equilibrium problem rather than a probabilistic learning task. Using a fixed 30-word vocabulary with explicit semantic clustering, we demonstrated that both related-word completion and next-word completion can be resolved through linear constraints and algebraic equilibrium, without training, iteration, or statistical inference.

Words were represented as fixed vector coordinates, not as learned embeddings. Tokenization was shown to act solely as a coordinate activation mechanism, while semantic structure was imposed explicitly through cluster-induced constraints. Within this setting, attention was redefined as structural compatibility under constraints, revealed by the equilibrium solution rather than computed through similarity scores or softmax normalization.

By explicitly constructing and displaying the Q, K, and V matrices, the work clarified that:

1) Q encodes what the input phrase demands,

2) K encodes which vocabulary elements are eligible to respond,

3) V maps the outcome directly back to vocabulary space.

The resulting attention pattern emerged deterministically as the set of variables that survive constraint enforcement. When multiple words satisfied the constraints equally, degeneracy was exposed rather than hidden, and a simple deterministic tie-breaking rule was applied for single-token output. The selection of "apple" in the illustrative example was therefore not a preference or likelihood estimate, but a reproducible consequence of equilibrium symmetry and ordering.

This explicit treatment highlights a fundamental distinction from transformer-based models: while conventional attention mechanisms rely on learned representations, stochastic optimization, and probabilistic normalization, the proposed framework treats language completion as a constrained equilibrium problem. Meaning emerges from structure, not from training trajectories.

The results suggest that next-word prediction does not inherently require probabilistic modeling or iterative optimization. Instead, for structured and interpretable settings, deterministic σ-regularized equilibrium offers a transparent, reproducible, and energy-efficient alternative. Future work will extend this formulation to larger vocabularies, richer constraint sets, and hybrid scenarios where deterministic equilibrium and data-driven components may coexist. The purpose of the reduced 8-token example is not empirical coverage but complete analytical transparency; larger vocabularies follow identically.

Finally, it is important to distinguish between mathematical scalability and practical deployment. While the σ-regularized

equilibrium formulation and the associated Q–K–V structure scale identically with vocabulary size, this does not imply that manual specification of semantic clusters and compatibility constraints is feasible for open-domain vocabularies containing tens of thousands of words. The explicit definition of structure introduces a well-known limitation commonly referred to as the *knowledge engineering bottleneck*. Accordingly, the proposed framework is not intended as a general-purpose replacement for large-scale language models, but as a deterministic inference mechanism for controlled, structured, or curated domains where semantic relations can be specified or externally supplied. Within this scope, the framework demonstrates that next-word resolution can be achieved without iterative learning or probabilistic optimization, emphasizing transparency, reproducibility, and analytical clarity over unrestricted coverage.

The deterministic equilibrium formulation intentionally exposes degeneracy when multiple vocabulary elements satisfy the imposed semantic constraints equally. In such cases, the equilibrium solution correctly reflects eligibility, not preference, likelihood, or ranking. The framework does not claim to distinguish among equally compatible outcomes in the absence of additional information. Consequently, any single-word output required for next-token completion is obtained through a deterministic post-processing rule applied after equilibrium resolution. The use of a lowest-index selection rule in the illustrative example is explicitly a sorting convention, not an intelligent choice mechanism, and does not reflect semantic preference or probability. This design choice is made for reproducibility and transparency. Unlike probabilistic models, where softmax normalization and numerical noise implicitly resolve degeneracy, the proposed framework makes equivalence classes explicit and separates structural eligibility from external selection logic. This distinction is intentional and highlights a fundamental difference between deterministic constraint-based inference and probabilistic ranking models.

In the context of this work, the term *"without iterative learning"* is used in a precise and restricted sense: it denotes the absence of statistical parameter fitting, gradient-based optimization, loss minimization, and iterative weight updates. The framework does not claim that semantic structure arises automatically or without human input. Instead, semantic relations are specified explicitly through deterministic constraints encoded in the Q and K matrices. This process is better characterized as structural specification or programming, rather than learning from data. The effort required to define such structure may, in practice, be comparable to or greater than data labeling, and this trade-off is acknowledged. The contribution of the present work is therefore not a reduction of human effort, but a demonstration that once structure is specified, inference can be performed deterministically, reproducibly, and without iterative optimization. This distinction separates the proposed framework from learning-based models while preserving conceptual clarity.

Abbreviations

A	Data Matrix
AI	Artificial Intelligence
ANN	Artificial Neural Network
CSP	Constraint Satisfaction Problem
GD	Gradient Descent
SGD	Stochastic Gradient Descent
d	Feature Dimension
Eanchor(k)	Anchor Energy of Block k
LLM	Large Language Model
Lσ	Anchor-Loss for σ-Regularized Solution
N	Number of Samples
Q–K	Query–Key
Q-K-V	Query–Key–Value
σ	Stabilizing Regularization Parameter
σ-Method	Deterministic σ-Regularized Learning Method
σ-March	Sequential σ-Stability Evaluation Process
σ-Block	Overlapping Deterministic Block
σ-Equilibrium	Unique Stationary Point of the σ-Regularized System
VSA	Vector Symbolic Architecture
W	Weight Vector
Wσ	σ-Regularized Deterministic Solution

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, L. and Polosukhin, I.,"Attention Is All You Need." *Advances in Neural Information Processing Systems*, 2017. https://doi.org/10.48550/arXiv.1706.03762

[2] Lin, T., Wang, Y., Liu, X. and Qiu, X. "A Survey of Transformers." *ACM Computing Surveys*, 2022. https://doi.org/10.1145/3505246

[3] Schwartz, R., Dodge, J., Smith, N. A. and Etzioni, O., "Green AI." *Communications of the ACM*, 2022. https://doi.org/10.1145/3381831

[4] Jain, S. and Wallace, B. "Attention Is Not Explanation." *NAACL*, 2022. https://doi.org/10.18653/v1/N19-1357

[5] A. N. Tikhonov, Solutions of Ill-Posed Problems, V. H. Winston & Sons, 1977.

[6] Cekirge, H. M. "Tuning the Training of Neural Networks by Using the Perturbation Technique." American Journal of Artificial Intelligence, 9(2), 107–109, 2025. https://doi.org/10.11648/j.ajai.20250902.11

[7] 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

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

[9] 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

[10] Cekirge, H. M. Cekirge_Perturbation_Report_v4. Zenodo, 2025. https://doi.org/10.5281/zenodo.17393651

[11] Cekirge, H. M. "Algebraic Cekirge Method for Deterministic and Energy-Efficient Transformer Language Models." American Journal of Artificial Intelligence, 9(2), 258–271, 2025. https://doi.org/10.11648/j.ajai.20250902.25

[12] Cekirge, H. M. "Deterministic σ-Regularized Benchmarking of the Cekirge Model Against GPT-Transformer Baseline." American Journal of Artificial Intelligence, 9(2), 272–280, 2025. https://doi.org/10.11648/j.ajai.20250902.26

[13] Cekirge, H. M. "The Cekirge Method for Machine Learning: A Deterministic σ-Regularized Analytical Solution for General Minimum Problems." American Journal of Artificial Intelligence, 9(2), 324–337, 2025. https://doi.org/10.11648/j.ajai.20250902.31

[14] Cekirge, H. M. "The Cekirge σ-Method in AI: Analysis and Broad Applications." *American Journal of Artificial Intelligence* 10(1), 14–33, 2026. https://doi.org/10.11648/j.ajai.20261001.12

[15] Pineau, J., Vincent-Lamarre, P., Sinha, K., Lariviere, V., Beygelzimer, A., d'Alche-Buc, F., Fox, E. and Larochelle, H., "Improving Reproducibility in Machine Learning Research." *JMLR*, 22(164):1–20, 2021. https://doi.org/10.5555/3455716.3455719

[16] Benning, M. and Burger, M. "Modern Regularization Methods." *Acta Numerica*, 2018 (still acceptable). https://doi.org/10.1017/S0962492918000016

[17] Tay, Y., Dehghani, M., Bahri, D. and Metzler, D., "Efficient Transformers: A Survey." *ACM CSUR*, 2023. https://doi.org/10.1145/3530811