2025. Volume 9, Number 1 Pages 75-94

UOT: 004.8
DOI: https://doi.org/10.30546/09090.2025.210.027

RESEARCH OF THE MEDICAL MULTI-AGENT DISTILLED LLM DIAGNOSTIC SYSTEMS USING GAME-THEORETIC OPTIMIZATION

JAVID ABBASLI[1*]

[1]*Azerbaijan Technical University,*

cavid.abbasli@aztu.edu.az

Baku, Azerbaijan

ARTICLE INFO

Article history:
Received:2025-10-01
Received in revised form:2025-10-02
Accepted:2025-10-15
Available online

Keywords:
multi-agent systems;
knowledge distillation;
game-theoretic optimization;
medical diagnostics;
large language models.
2010 Mathematics Subject Classifications: 91A80, 90C70, 68T05, 68T07, 68T20

ABSTRACT

This study presents a multi-agent framework for medical diagnostics by integrating knowledge distillation with game-theoretic optimization. Three specialized teacher models are trained and distilled into lightweight student agents using temperature-scaled soft labels (α=0.7, T=3.0), achieving 6.5× compression. The system employs a utility function incorporating accuracy, confidence, consensus, and time penalty to dynamically select the optimal agent via Nash equilibrium for each case. Unlike ensemble methods relying on simple voting, our approach maximizes contextual utility through strategic agent coordination. Experimental results demonstrate that the multi-agent system significantly outperforms individual agents, achieving substantial accuracy improvements. This framework enables efficient medical decision support on resource-constrained devices while maintaining robust diagnostic performance through collaborative artificial intelligence.

1. Introduction

The rapid advancement of artificial intelligence has revolutionized medical diagnostics, with large language models (LLMs) demonstrating remarkable capabilities in clinical decision support and disease classification [6, 16]. However, deploying these models in real-world medical settings faces two fundamental challenges: computational efficiency and diagnostic reliability.

State-of-the-art language models such as GPT-4 [15], BERT [3], and their medical variants like BioBERT [12] and ClinicalBERT [1] typically contain hundreds of millions to billions of parameters, requiring substantial computational resources that make them impractical for resource-constrained environments. These limitations severely restrict their application in mobile health systems, rural clinics, point-of-care diagnostics, and developing countries where computational infrastructure is limited [23]. The computational burden introduces latency that can be unacceptable in emergency scenarios where immediate decisions are critical.

Beyond computational constraints, medical diagnostics involves high-stakes decision-making where erroneous diagnoses can have life-threatening consequences. While individual AI models

show promising performance in controlled settings, they often lack the robustness required for clinical deployment [20]. The medical community has long recognized that collaborative decision-making—where multiple expert opinions are considered—leads to more accurate diagnoses than relying on a single practitioner [11]. However, traditional ensemble methods that simply aggregate predictions through voting or averaging fail to capture the nuanced decision-making processes used by human medical teams [4].

Knowledge distillation, introduced by Hinton et al. [8], offers a promising solution to computational challenges by transferring knowledge from large teacher models to smaller student models. By training students to mimic soft probability distributions rather than just hard labels, significant reductions in model size can be achieved while preserving performance [7]. Temperature scaling smooths the teacher's output distribution, revealing valuable inter-class relationships [8]. However, distilled models inevitably experience performance degradation, with typical accuracy retention rates of 70-90% depending on compression ratio [17, 9]. This performance gap raises concerns in critical medical applications.

To address both challenges simultaneously, this study proposes a novel framework combining knowledge distillation with multi-agent systems and game-theoretic optimization. Rather than relying on a single distilled model, we create multiple specialized lightweight agents, each distilled from teacher models with different initializations. These agents collaborate through a game-theoretic mechanism to select the most appropriate diagnosis based on a utility function incorporating accuracy, confidence, consensus, and response time. Game theory provides a rigorous framework for modeling strategic interactions among rational decision-makers [22]. By formulating agent selection as a Nash equilibrium problem [14], we ensure optimal strategic choice given all agents' capabilities and predictions. This differs fundamentally from fixed-weight ensemble methods, adapting dynamically to each diagnostic case [2, 24].

The framework aligns naturally with medical specialization practices. Just as physicians specialize in cardiology, infectious diseases, or general practice, our system trains separate teacher models that develop implicit specializations during training. Each teacher is distilled into a lightweight agent inheriting domain-relevant capabilities while maintaining efficiency. The utility function combines four clinically relevant components weighted according to medical priorities, with diagnostic accuracy receiving highest weight as it directly impacts patient outcomes.

This research makes several significant contributions. First, we demonstrate that multi-agent collaboration through game-theoretic optimization substantially exceeds individual distilled model performance. While individual agents achieve 68-74% validation accuracy, the collaborative system reaches 93.5% test accuracy—a +23.5 percentage point improvement over single-agent baseline. Second, we introduce a principled game-theoretic framework incorporating multiple decision factors through a utility function. Third, comprehensive ablation studies demonstrate each component's contribution, with consensus removal causing the largest performance drop. Fourth, we achieve 6.5× parameter reduction (573K → 88.6K per agent) while maintaining competitive accuracy, enabling deployment on resource-constrained devices. The multi-agent system requires only 265,758 total parameters (still 2.16× smaller than a single teacher model) yet delivers performance far exceeding any individual agent. Agent selection analysis reveals balanced diversification with no dominant agent, indicating the system successfully leverages complementary strengths across different diagnostic scenarios. The

framework naturally aligns with medical specialization practices where human physicians consult appropriate specialists based on case characteristics, but implements this principle automatically through utility maximization without requiring explicit domain labels or manual expertise specification.

The remainder of this paper is organized as follows. Section 2 reviews related work in knowledge distillation, multi-agent systems, and medical AI. Section 3 presents the mathematical framework of knowledge distillation, including transformer architecture [21], temperature scaling, and loss functions with Adam optimizer [10]. Section 4 introduces the game-theoretic multi-agent framework, defining utility function components and Nash equilibrium formulation. Section 5 describes experimental methodology, including dataset construction, model architectures, and evaluation metrics and presents comprehensive results including teacher training, distillation performance, multi-agent outcomes, and ablation studies. Section 6 discusses implications for clinical and concludes with key contributions and future research directions.

2. Related works

The intersection of knowledge distillation, multi-agent systems, and medical artificial intelligence represents a rapidly evolving research frontier. This section examines prior work across these domains and positions our contributions within the broader landscape. The concept of knowledge distillation was pioneered by Hinton et al. [8], who demonstrated that smaller student networks could learn to approximate the behavior of larger teacher networks by matching their soft output distributions. The key insight was that the teacher's softened probabilities, obtained through temperature scaling, contain richer information about inter-class similarities than hard one-hot labels. This approach has been successfully applied to compress large-scale models while maintaining competitive performance. Sanh et al. [17] introduced DistilBERT, which retained 97% of BERT's language understanding capabilities while reducing model size by 40% and improving inference speed by 60%. Jiao et al. [9] further advanced this work with TinyBERT, achieving 7.5× compression through a two-stage distillation process that transfers knowledge from both the intermediate layers and the output layer. Gou et al. [7] provided a comprehensive survey categorizing distillation methods into response-based, feature-based, and relation-based approaches. Despite these advances, most distillation work focuses on single student models, and the performance gap between teacher and student remains a fundamental challenge, particularly in high-stakes applications where even small accuracy losses are unacceptable.

Medical applications of large language models. The application of LLMs to medical domains has garnered significant attention following breakthroughs in general-purpose language understanding. Esteva et al. [6] demonstrated that deep learning models could achieve dermatologist-level accuracy in skin cancer classification, marking an early success in medical AI. Rajkomar et al. [16] discussed the broader potential of machine learning in medicine, highlighting both opportunities and challenges including interpretability, fairness, and clinical integration. Recent work by Singhal et al. [18] showed that large language models can encode substantial clinical knowledge, achieving expert-level performance on medical licensing exam questions. Thirunavukarasu et al. [19] provided a comprehensive review of LLMs in medicine, discussing applications in clinical documentation, medical education, and diagnostic support while emphasizing the need for rigorous validation before clinical deployment. However, these

powerful models typically require substantial computational resources. Lee et al. [12] developed BioBERT by pre-training on biomedical corpora, while Alsentzer et al. [1] created ClinicalBERT using electronic health records, but both retain the computational burden of BERT-scale architectures. The tension between model capability and deployment feasibility remains unresolved in medical AI applications.

Ensemble methods and multi-agent systems. The principle that combining multiple models often outperforms individual models has been well-established in machine learning. Dietterich [4] provided foundational work on ensemble methods, demonstrating that diversity among base learners is crucial for ensemble performance. In medical contexts, Kurvers et al. [11] showed that pooling independent diagnostic judgments from multiple clinicians significantly improves accuracy, echoing the clinical practice of seeking second opinions for difficult cases. However, traditional ensemble methods such as majority voting, bagging, and boosting treat all models equally or assign fixed weights regardless of input characteristics. Multi-agent reinforcement learning, surveyed by Busoniu et al. [2] and Zhang et al. [24], has explored more sophisticated coordination mechanisms, but these approaches typically focus on sequential decision-making in controlled environments rather than one-shot classification tasks. The application of game-theoretic principles to model selection in medical diagnostics remains largely unexplored, representing a significant gap in the literature.

Game theory in artificial intelligence. Game theory, formalized by Von Neumann and Morgenstern [22], provides mathematical tools for analyzing strategic interactions among rational agents. Nash's seminal work [14] on non-cooperative games established the concept of Nash equilibrium, where no player can unilaterally improve their outcome by changing strategy. While game-theoretic approaches have been extensively applied in multi-agent reinforcement learning and algorithmic game theory, their application to model selection and ensemble construction remains limited. Most existing work treats ensemble combination as a supervised learning problem with fixed weights, rather than as a strategic game where each model's contribution depends on the predictions and confidences of other models. Our work bridges this gap by formulating agent selection as a utility maximization problem under Nash equilibrium constraints.

Model compression for resource-constrained deployment. The deployment of AI models in resource-constrained environments has motivated extensive research in model compression. Beyond knowledge distillation, techniques include pruning, quantization, and neural architecture search. Wahl et al. [23] highlighted the critical need for efficient AI systems in resource-poor healthcare settings, where computational infrastructure is limited but medical needs are high. Dosovitskiy et al. [5] demonstrated that transformer architectures, originally designed for natural language processing [21], could be effectively applied to vision tasks, expanding the scope of potential compression targets. However, most compression research evaluates models in isolation, without considering how multiple compressed models might collaborate to recover the performance of their larger predecessors. Our multi-agent framework addresses this limitation by showing that strategic coordination among distilled models can substantially exceed individual model performance.

Gaps and contributions. Despite progress in each of these areas individually, several important gaps remain. First, knowledge distillation research has not adequately addressed how multiple distilled models with complementary specializations might collaborate to overcome individual

performance limitations. Second, medical AI applications have not fully leveraged game-theoretic principles for model coordination despite the natural parallel to clinical team decision-making. Third, ensemble methods lack principled approaches for dynamic, context-dependent model selection that adapts to each input case. Our work addresses these gaps by introducing a game-theoretic multi-agent framework that combines knowledge distillation with Nash equilibrium-based agent selection, demonstrating that strategic collaboration among lightweight models can achieve performance substantially exceeding individual agents while maintaining computational efficiency suitable for resource-constrained deployment.

3. Mathematical model of the knowledge distillation process for medical large language models

Knowledge distillation provides a principled approach to compress large neural networks into smaller, computationally efficient models while preserving much of their predictive performance. This section presents the mathematical foundations underlying our distillation methodology, beginning with the transformer architecture [21] and progressing through the complete distillation framework including temperature scaling, loss function design, and optimization procedures. We provide rigorous mathematical definitions for all components, specify the properties of the spaces involved, and explain the rationale behind each architectural choice.

Medical Diagnostic Function Formulation. The medical diagnostic task is formalized as a deterministic mapping from a discrete input space to a discrete output space augmented with a continuous confidence measure. Let $X \subset N^n$ denote the space of tokenized patient complaints, where each input sequence consists of $n = 32$ tokens drawn from a finite vocabulary of size $V = 200$. More precisely, $X = 1,2,\dots,V^n$ is the Cartesian product space of all possible token sequences of fixed length n. This space is finite with cardinality $|X| = V^n$, discrete (no notion of continuity between sequences), and equipped with the discrete topology. Each element $x \in X$ represents a patient complaint encoded as an ordered sequence of integer token indices: $x = (x_1, x_2, \dots, x_n)$ where $x_i \in 1,2,\dots,V$ for all i.

The diagnostic label space is defined as $\mathcal{D} = d_1, d_2, \dots, d_K$, a finite discrete set containing $K = 10$ possible diagnostic categories. In our experimental implementation, these ten categories represent distinct disease classes constructed through our synthetic pattern-based dataset, where each class is associated with specific keyword patterns. The space $\mathcal{D}$ is equipped with the discrete metric $\rho(d_i, d_j) = 0$ if $i = j$ and $\rho(d_i, d_j) = 1$ if $i \neq j$, forming a discrete metric space $(\mathcal{D}, \rho)$. The confidence space is the unit interval $[0,1] \subset R$, a compact and connected subset of the real line equipped with the standard Euclidean metric inherited from R.

The diagnostic function is formally defined as:

$$f: X \to D \times [0,1] \qquad (1)$$

This function maps each tokenized input sequence $x \in X$ to an ordered pair $(\hat{d}, c)$ where $\hat{d} \in \mathcal{D}$ represents the predicted diagnosis and $c \in [0,1]$ quantifies the model's confidence in this prediction. The confidence value c is interpreted as the maximum probability assigned by the model's output softmax distribution, serving as a scalar measure of prediction certainty. Mathematically, if $p \in R^K$ denotes the probability vector over all K diagnoses (defined rigorously below), then $c = \max_{k=1,\dots,K} p_k$ and $\hat{d} = d_{arg\max_k p_k}$. This formulation explicitly captures diagnostic uncertainty, which is clinically crucial as physicians must quantify their confidence when making diagnoses under incomplete information.

Additionally, modern large language models are built upon the transformer architecture [21], which processes sequential input through multiple layers of self-attention and feed-forward operations. The transformer maps discrete token sequences into continuous vector representations in high-dimensional Euclidean space, enabling gradient-based optimization. Our teacher model consists of $L_T = 4$ transformer encoder layers, each performing nonlinear transformations on vector representations.

The architecture begins with an embedding layer that maps discrete tokens to continuous vectors. Formally, the embedding is a linear map $E: 1,2,\dots,V \to R^d$ implemented as a lookup table, where $V = 200$ is the vocabulary size and $d = 128$ is the embedding dimension. Each token index $x_i \in 1,2,\dots,V$ is mapped to a learned vector $E(x_i) \in R^d$. For a sequence $x = (x_1, x_2, \dots, x_n)$, the embedding operation produces a matrix:

$$h_0 = Embedding(x) \in R^{n \times d} \quad (2)$$

where the *i-th* row of h_0 is $E(x_i)$. Here $Embedding(\cdot)$ denotes the composition of the lookup operation followed by positional encoding addition. Specifically, $h_0 = [E(x_1); E(x_2); \dots; E(x_n)] + P$ where $P \in R^{n \times d}$ is a learned positional encoding matrix that provides position information to the otherwise permutation-invariant attention mechanism. The space $R^{n \times d}$ is a finite-dimensional real Hilbert space with inner product $\langle A, B \rangle = \mathrm{tr}(A^T B)$ and induced Frobenius norm $\| A \|_F = \sqrt{\mathrm{tr}(A^T A)}$.

Each transformer layer $\ell \in 1,2,\dots,L_T$ applies a sequence of operations mapping $R^{n \times d}$ to itself. The layer transformation is denoted:

$$h_\ell = Transformer_\ell(h_{\ell-1}) \quad (3)$$

where $h_\ell \in R^{n \times d}$ represents the hidden state matrix at layer ℓ. Each row of h_ℓ corresponds to the d-dimensional representation of one token position. The subscript ℓ distinguishes different layer instances, each with independent learnable parameters. The function $Transformer_\ell: R^{n \times d} \to R^{n \times d}$ is a composition of two main components: multi-head self-attention followed by a position-wise feed-forward network, each wrapped with residual connections and layer normalization [21].

The multi-head self-attention mechanism computes weighted combinations of token representations, allowing each position to attend to all other positions. Let $h \in R^{n \times d}$ denote the input to the attention layer. The attention mechanism uses three learnable linear projections: query $W^Q \in R^{d \times d}$, key $W^K \in R^{d \times d}$, and value $W^V \in R^{d \times d}$. These matrices project the input into query, key, and value spaces:

$$Q = hW^Q, \quad K = hW^K, \quad V = hW^V \quad (4)$$

where $Q, K, V \in R^{n \times d}$ are the query, key, and value matrices respectively. Each row of Q represents the query vector for one token position, each row of K represents the key vector, and each row of V represents the value vector. The matrices W^Q, W^K, W^V are learnable parameters that transform the input representations into specialized spaces for computing attention.

The multi-head attention splits the d-dimensional space into $h = 4$ independent subspaces, each of dimension $d_k = d/h = 32$. Each attention head operates independently on its subspace. The multi-head output is computed as:

$$MultiHead(Q, K, V) = Concat(head_1, head_2, \dots, head_h)W^O \quad (5)$$

where $Concat(\cdot)$ denotes concatenation along the feature dimension, and $W^O \in R^{d\times d}$ is an output projection matrix. Each attention head $i \in 1,2,\dots,h$ computes:

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \qquad (6)$$

where $W_i^Q, W_i^K, W_i^V \in R^{d\times d_k}$ are head-specific projection matrices that extract the i-th subspace. The term "head" refers to an independent attention computation unit, and "multi-head" indicates parallel computation across multiple such units. This architecture allows the model to attend to different aspects of the input simultaneously. The core attention function implements scaled dot-product attention:

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (7)$$

This function takes three matrices as input: queries $Q \in R^{n\times d_k}$, keys $K \in R^{n\times d_k}$, and values $V \in R^{n\times d_k}$. The operation $QK^T \in R^{n\times n}$ computes pairwise dot products between all query and key vectors, measuring their similarity. The (i,j) entry of QK^T equals $Q_{i,:} \cdot K_{j,:}^T$, representing the affinity between token i (as query) and token j (as key). The scaling factor $1/\sqrt{d_k}$ with $d_k = 32$ normalizes these dot products to have approximately unit variance, preventing extremely large values that would cause the subsequent softmax to produce near one-hot distributions with vanishing gradients. The softmax function is applied row-wise, converting each row of the scaled similarity matrix into a probability distribution over all token positions. Finally, multiplication by V computes weighted averages of value vectors according to these attention weights.

Following the attention sublayer, each transformer layer applies a position-wise feed-forward network (FFN). This network processes each token position independently using the same parameters across all positions. The FFN is defined as:

$$FFN(x) = max(0, xW_1 + b_1)\,W_2 + b_2 \quad (8)$$

where $x \in R^d$is the input vector for a single position, $W_1 \in R^{d\times d_{\text{ff}}}$ and $W_2 \in R^{d_{\text{ff}}\times d}$ are weight matrices, and $b_1 \in R^{d_{\text{ff}}}$and $b_2 \in R^d$ are bias vectors.

In our teacher model, the intermediate dimension is $d_{\text{ff}} = 2d = 256$. The notation $max(0,\cdot)$ denotes the rectified linear unit (ReLU) activation function applied element-wise, introducing non-linearity: $\text{ReLU}(z) = z$ if $z > 0$ and $\text{ReLU}(z) = 0$ if $z \le 0$. This two-layer network with ReLU activation enables the model to learn complex nonlinear transformations of token representations. The expansion to $d_{\text{ff}} = 256$ dimensions allows the network to compute richer intermediate features before projecting back to the $d = 128$ dimensional output space.

After $L_T = 4$ transformer layers, the final hidden state matrix $h_L \in R^{n\times d}$ encodes contextual information about the entire input sequence. To obtain a fixed-size sequence-level representation suitable for classification, we apply mean pooling across the sequence dimension:

$\bar{h} = \frac{1}{n}\sum_{i=1}^{n} h_{L,i,:} \in R^d$ (9) where $h_{L,i,:}$ denotes the *i-th* row of h_L (the representation of the *i-th* token position). This operation computes the arithmetic mean of all token representations, producing a single vector h that aggregates information from the entire sequence. Alternative pooling strategies include taking only the first token representation (as in BERT [3]) or *max pooling*, but mean pooling provides a simple and effective aggregation in our setting.

The pooled representation h is then projected to the output space through an affine transformation:

$$z = W_{\text{out}}\bar{h} + b_{\text{out}} \qquad (10)$$

where $W_{\text{out}} \in R^{K\times d}$ is the output weight matrix, $b_{\text{out}} \in R^K$ is the output bias vector, and $z \in R^K$ is the logit vector. The term "logit" refers to the unnormalized log-odds of each class before applying the softmax function. Each component z_k for $k \in 1,2,\dots,K$ represents the model's raw score for diagnosis d_k, with higher values indicating stronger evidence for that diagnosis. These logits are real numbers without bound: $z_k \in R$.

Finally, the logits are converted to a probability distribution over diagnoses using the softmax function:

$$p = \text{softmax}(z) \in R^K \quad (11)$$

where the softmax function is defined component-wise as:

$$p_k = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_k}}, \quad k = 1,2,\dots,K \quad (12)$$

The softmax function $\sigma: R^K \to \Delta^{K-1}$ maps the unbounded logit vector to the $(K-1)$-dimensional probability simplex $\Delta^{K-1} = \{p \in R^K : p_k \geq 0, \sum_k p_k = 1\}$. This ensures that the output satisfies the axioms of probability: non-negativity ($p_k \geq 0$ for all k) and normalization ($\sum_k p_k = 1$). The exponential function ensures positivity, and the normalization by the sum ensures the probabilities sum to one. The softmax function is smooth (infinitely differentiable), monotonically preserves the ordering of logits, and has well-defined gradients facilitating backpropagation [21]. We use softmax rather than alternatives like sigmoid or direct normalization because it provides a proper probabilistic interpretation and has favorable gradient properties for multi-class classification.

On the model size analysis side, the computational and memory requirements of transformer models scale with the number of learnable parameters. We derive the parameter count by summing contributions from all components. Each transformer layer contains four main parameter groups: the query, key, value, and output projections in the attention mechanism (each $d \times d$ matrices, contributing $4d^2$ parameters per layer), the two feed-forward weight matrices ($d \times d_{\text{ff}}$ and $d_{\text{ff}} \times d$, contributing $2d \cdot d_{\text{ff}}$ parameters), feed-forward bias vectors (contributing $d_{\text{ff}} + d$ parameters), and layer normalization parameters (two sets of d parameters each for scale and shift, contributing $2d$ parameters). The total parameter count for a transformer with L layers, embedding dimension d, and feed-forward dimension d_{ff} is:

$$N = L \times (4d^2 + 2d \cdot d_{\text{ff}} + d_{\text{ff}} + d + 2d) + Vd + Kd \quad (13)$$

where the terms Vd and Kd account for the embedding lookup table ($V \times d$) and output projection ($K \times d$) respectively. For our teacher model with $L_T = 4, d_T = 128, d_{\text{ff}} = 256, V = 200$, and $K=10$, we can compute:

$$N_T = 4 \times (4 \times 128^2 + 2 \times 128 \times 256 + 256 + 128 + 256) + 200 \times 128 + 10 \times 128 \quad (14)$$

$$N_T = 4 \times (65536 + 65536 + 640) + 25600 + 1280 = 4 \times 131712 + 26880 = 553728 \qquad (15)$$

A more precise count considering all bias terms and layer normalization gives $N_T = 573194$ parameters as observed in our implementation. Similarly, the student model with $L_S = 2, d_S = 64, d_{\text{ff}}^S = 128$ contains:

$$N_S = 2 \times (4 \times 64^2 + 2 \times 64 \times 128 + 128 + 64 + 128) + 200 \times 64 + 10 \times 64 (16)$$

yielding $N_S = 88586$ parameters. The compression ratio, defined as the ratio of teacher to student parameter counts, is:

$$r_c = \frac{N_T}{N_S} = \frac{573194}{88586} \approx 6.47 \qquad (17)$$

This ratio quantifies the reduction in model size achieved through distillation. A compression ratio of $r_c \approx 6.5$ means the student requires approximately $1/6.5 \approx 15.4\%$ of the teacher's parameters, directly translating to proportional reductions in memory footprint (for storing model weights), inference time (fewer operations), and energy consumption (fewer computations). Knowledge distillation [8] aims to train a compact student model to approximate a large pre-trained teacher model's behavior.

Let $f_T: X \to R^K$ denote the teacher model mapping inputs to logit vectors, parameterized by $\theta_T \in R^{N_T}$ where each component of θ_T represents one learnable parameter (weights and biases). The teacher is pre-trained on a dataset $\mathcal{D} = \left(x^{(n)}, y^{(n)}\right)_{n=1}^{N}$ consisting of $N = 1000$ training samples, where each $x^{(n)} \in X$ is a tokenized input sequence and each $y^{(n)} \in 0,1^K$ is a one-hot encoded label vector with $y_k^{(n)} = 1$ if sample n belongs to class k and $y_k^{(n)} = 0$ otherwise.

The distillation objective is to find student parameters $\theta_S \in R^{N_S}$ that minimize the expected distillation loss over the training distribution:

$$\theta_S^* = \arg\min_{\theta_S} E_{(x,y)\sim\mathbb{D}} \left[L_{\text{distill}}(f_S(x;\theta_S), f_T(x;\theta_T), y)\right] (18)$$

$$\text{subject to} \quad |\theta_S|_0 \leq N_S \ (19)$$

where $|\theta_S|_0$ denotes the ℓ_0 quasi-norm counting the number of non-zero parameters, ensuring the student respects the target model size. The expectation $E_{(x,y)\sim\mathbb{D}}[\cdot]$ denotes the average over the empirical data distribution, which in practice is approximated through mini-batch stochastic gradient descent.

The loss function $L_{\text{distill}}: R^K \times R^K \times 0,1^K \to R_{\geq\mathbb{0}}$ measures the discrepancy between student predictions, teacher predictions, and ground truth, designed to transfer both the teacher's correct predictions and its learned inter-class relationships to the student.

When teacher models achieve very high accuracy (100% in our experiments), their output distributions become extremely peaked with the correct class receiving probability near 1 and all other classes receiving probabilities near 0. This concentration of probability mass on a single class provides minimal information about the teacher's learned similarities between classes. For example, if a teacher predicts class 1 with probability 0.999 and distributes the remaining 0.001 among classes 2-10, a student learning only from these probabilities cannot discern whether classes 2 and 3 are more similar to class 1 than classes 8 and 9.

Temperature scaling [8] addresses this by introducing a temperature parameter T>0 that controls the entropy of the output distribution. For a logit vector z\in\mathbb{R}^K, the temperature-scaled probability for class i is defined as:

$$q_i(T) = \frac{e^{\frac{z_i}{T}}}{\sum_{j=1}^{K} e^{\frac{z_j}{T}}}, i = 1,2,\ldots,K \quad (20)$$

Here $q_i(T) \in [0,1]$ denotes the probability assigned to class i under temperature T, and the vector $q(T) = \left(q_1(T), q_2(T), \ldots, q_K(T)\right)^T \in \Delta^{K-1}$ forms a valid probability distribution satisfying

$\sum_{i=1}^{K} q_i(T) = 1$ and $q_i(T) \geq 0$ for all i. The notation $q_i(T)$ explicitly shows the dependence on temperature, distinguishing it from the standard softmax probabilities p_i computed at $T = 1$.

The parameter T controls the smoothness of the distribution. When $T = 1$, equation (20) reduces to the standard softmax : $q_i(1) = p_i$. When $T > 1$, dividing logits by T before exponentiation reduces their magnitude, causing the exponential to vary more slowly and producing a smoother, higher-entropy distribution. The Shannon entropy of the temperature-scaled distribution is:

$$H\big(q(T)\big) = -\sum_{i=1}^{K} q_i(T) \ln q_i\,(T) \qquad (21)$$

which increases monotonically with T. As $T \to \infty$, all logits become approximately equal after scaling, yielding $q_i(T) \to 1/K$ (uniform distribution) with maximum entropy $H = lnK$. Conversely, when $0 < T < 1$, logits are amplified, creating a sharper, lower-entropy distribution.

In our experiments, we use $T = 3.0$, which substantially smooths the teacher's output distribution while maintaining meaningful distinctions between classes. The choice of temperature involves a fundamental trade-off. If T is too small ($T \approx 1$), the soft targets remain nearly one-hot and provide little additional information beyond hard labels, defeating the purpose of distillation. If T is too large ($T \gg 10$), the soft targets become nearly uniform, losing discriminative information about which classes the teacher considers more or less plausible. Empirical studies [8] suggest $T \in [2,5]$ works well across many tasks. We selected $T = 3.0$ based on preliminary experiments showing it provides good balance for our dataset.

The distillation loss combines two objectives: matching ground-truth labels (supervised learning) and matching teacher predictions (knowledge transfer). The composite loss is:

$$L_{\text{distill}}(\theta_S) = \alpha \cdot L_{\text{hard}}(\theta_S) + (1 - \alpha) \cdot L_{\text{soft}}(\theta_S, \theta_T) \qquad (22)$$

where $\alpha \in [0,1]$ is a weighting hyperparameter. We use $\alpha = 0.7$, placing 70% weight on matching ground truth and 30% on matching teacher outputs. This balance ensures the student learns correct classifications while benefiting from the teacher's soft label structure.

The hard loss is the standard cross-entropy between student predictions and one-hot ground-truth labels:

$$L_{\text{hard}}(\theta_S) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} y_k^{(n)} \log p_S^{(n)}\,(k) \qquad (23)$$

where N is the batch size, $y_k^{(n)} \in 0,1$ is the ground-truth label (1 if sample n belongs to class k, 0 otherwise), and $p_S^{(n)}(k) \in [0,1]$ is the student's predicted probability for class k on sample n computed via standard softmax (temperature $T = 1$). Since $y^{(n)}$ is one-hot, only one term in the inner sum is non-zero: if sample n has true class k^*, then $L_{\text{hard}} = -\frac{1}{N}\sum_{n=1}^{N} log\, p_S^{(n)}(k^*)$. This loss is minimized when the student assigns high probability to the correct class.

The soft loss measures the Kullback-Leibler (KL) divergence between temperature-scaled distributions of the teacher and student:

$$L_{\text{soft}}(\theta_S, \theta_T) = T^2 \cdot \frac{1}{N}\sum_{n=1}^{N} \text{KL}\left(q_T^{(n)}(T) \parallel q_S^{(n)}(T)\right) \qquad (24)$$

where $q_T^{(n)}(T) \in \Delta^{K-1}$ and $q_S^{(n)}(T) \in \Delta^{K-1}$ are the temperature-scaled probability distributions for sample n from teacher and student respectively, computed via equation (20). The KL divergence is defined as:

$$\mathrm{KL}(p \parallel q) = \sum_{k=1}^{K} p_k \log\left(\frac{p_k}{q_k}\right) \quad (25)$$

The KL divergence is a non-negative measure of distributional dissimilarity: $\mathrm{KL}(p \parallel q) \geq 0$ with equality if and only if $p = q$. It is not symmetric $\left(\mathrm{KL}(p \parallel q) \neq \mathrm{KL}(q \parallel p)\right)$ and thus not a true distance metric, but it serves as an effective loss function for matching distributions. The soft loss encourages the student's softened predictions to match the teacher's softened predictions, transferring knowledge about inter-class similarities.

The T^2 coefficient in equation (24) is critical for maintaining stable gradients across different temperature values. To understand this scaling, consider the gradient of the temperature-scaled softmax with respect to logit z_j:

$$\frac{\partial q_i(T)}{\partial z_j} = \frac{1}{T} q_i(T)\left(\delta_{ij} - q_j(T)\right) \quad (26)$$

where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise is the Kronecker delta. The $1/T$ factor indicates that gradients scale inversely with temperature. When computing gradients of the KL loss via backpropagation, this $1/T$ factor appears twice (once for the student's softmax gradient and once for the logarithm's argument gradient), resulting in an overall $1/T^2$ scaling. Multiplying the loss by T^2 exactly compensates this effect, ensuring gradient magnitudes remain consistent regardless of temperature choice. This stabilization is essential for combining losses at different temperatures or for tuning T without retuning learning rates. We train the student model using the Adam optimizer [10], an adaptive learning rate method that maintains per-parameter learning rates based on first and second moment estimates of gradients. At training iteration t, we compute the gradient of the distillation loss with respect to student parameters:

$$g_t = \nabla_{\theta_S} L_{\text{distill}}\left(\theta_{S,t}\right) \in R^{N_S} \quad (27)$$

This gradient vector indicates the direction of steepest ascent in the loss landscape, with each component specifying how much to adjust the corresponding parameter to increase the loss (we will move opposite this direction to decrease the loss). For this situation, ADAM maintains two moving average estimates. The first moment estimate (mean of gradients) is updated as:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (28)$$

where $\beta_1 = 0.9$ is the exponential decay rate for the first moment. This gives approximately equal weight to gradients from the last $1/(1 - \beta_1) = 10$ iterations. The second moment estimate (uncentered variance of gradients) is:

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (29)$$

where $\beta_2 = 0.999$ is the decay rate for the second moment, and g_t^2 denotes element-wise squaring. This tracks the magnitude of typical gradient values over approximately $1/(1 - \beta_2) = 1000$ iterations.

Both m_t and v_t are initialized to zero vectors, which creates a bias toward zero in early iterations. Adam corrects this initialization bias by computing:

$$\widehat{m_t} = \frac{m_t}{1-\beta_1^t}, \quad \widehat{v_t} = \frac{v_t}{1-\beta_2^t} \quad (30)$$

where the denominators $1 - \beta_1^t$ and $1 - \beta_2^t$ approach 1 as t increases, eliminating the correction asymptotically while ensuring accurate estimates in early training. Finally, parameters are updated via:

$$\theta_{S,t+1} = \theta_{S,t} - \eta \frac{\widehat{m_t}}{\sqrt{\widehat{v_t}}+\varepsilon} \quad (31)$$

where $\eta = 3 \times 10^{-5}$ is the global learning rate and $\varepsilon = 10^{-8}$ is a small constant preventing division by zero. The update step size for each parameter is $\eta \widehat{m}_{t,i} / \left(\sqrt{\widehat{v}_{t,i}} + \varepsilon\right)$, which is proportional to the estimated mean gradient magnitude $\widehat{m}_{t,i}$ and inversely proportional to the square root of the estimated variance $\widehat{v}_{t,i}$. This adaptive scaling ensures that parameters with consistently large gradients receive smaller effective learning rates (preventing instability), while parameters with small gradients receive larger effective learning rates (accelerating convergence).

We use a learning rate $\eta = 3 \times 10^{-5}$ for student training, which is 10× smaller than the teacher's learning rate $\eta_T = 3 \times 10^{-4}$. This smaller rate reflects the different optimization landscape: the teacher learns from scratch to fit data, while the student learns to match a fixed teacher's behavior. Smaller learning rates prevent the student from diverging from the teacher's soft targets during training. Training proceeds for 20 epochs with batch size 32, gradient clipping at maximum norm 1.0 to prevent exploding gradients, and weight decay $\lambda = 10^{-5}$ for $ridge\ resssion$.

4. Game-theoretic optimization for medical multi-agent distilled large language models

The distillation process described in Section 3 produces compact student models that retain much of their teacher's diagnostic capability while requiring substantially fewer computational resources. However, individual distilled models inevitably suffer performance degradation compared to their teachers, with our experiments showing approximately 30 percentage point accuracy drops from perfect teacher performance to 70% student accuracy. This degradation raises concerns about reliability in high-stakes medical applications where diagnostic errors can have serious consequences. This section introduces a multi-agent framework that addresses this limitation through strategic collaboration among multiple distilled models, leveraging game-theoretic principles to recover and exceed individual agent performance.

We construct a system consisting of three distilled agents, denoted $\mathcal{A} = \{a_1, a_2, a_3\}$, each obtained through the knowledge distillation procedure outlined in Section 3. Agent a_i for $i \in \{1,2,3\}$ is parameterized by $\theta_{S,i} \in \mathbb{R}^{N_S}$ where $N_S = 88{,}586$ as computed in equation (16). Although all three agents share identical architectural specifications (2 transformer layers, 64-dimensional embeddings, 4 attention heads), they are distilled from different teacher models trained with different random initialization seeds (100, 200, 300). This diversity in initialization leads to different optimization trajectories during teacher training, causing each teacher to develop subtly different decision boundaries and feature representations. These differences propagate through distillation, resulting in student agents with complementary strengths and weaknesses despite their architectural uniformity.

Each agent operates independently during the forward pass. Given an input sequence $x \in X$, agent a_i computes its diagnostic prediction through the transformer architecture described in equations (2)-(12), producing a logit vector $z_i = f_i(x; \theta_{S,i}) \in \mathbb{R}^K$ where f_i denotes the forward function of agent i. From these logits, the agent derives three clinically relevant quantities. First, the predicted diagnosis is the class with maximum probability: $d_i = d_{\text{argmax}_k p_{i,k}}$ where $p_i = \text{softmax}(z_i)$ is the probability distribution over diagnoses. Second, the confidence score is the maximum probability: $c_i = \max_{k=1,\ldots,K} p_{i,k} \in [0,1]$, quantifying the agent's certainty about its

prediction. Third, the response time $t_i \in \mathbb{R}_+$ measures the elapsed time (in seconds) from input reception to output generation. The agent's complete response is thus the triple:

$$r_i = (d_i, c_i, t_i) \in \mathcal{D} \times [0,1] \times R_+ \qquad (32)$$

For a given input x, the multi-agent system collects responses from all three agents in parallel, forming the response set:

$$\mathcal{R}(x) = \{r_1, r_2, r_3\} = \{(d_1, c_1, t_1), (d_2, c_2, t_2), (d_3, c_3, t_3)\} \qquad (33)$$

The main question is which agent's prediction the system should trust when it receives three potentially conflicting diagnoses with different confidence levels. Traditional ensemble methods such as majority voting (select the diagnosis predicted by most agents) or confidence-weighted averaging fail to account for the contextual nature of agent expertise and the multi-dimensional nature of diagnostic quality. Our approach formulates this selection problem as a game-theoretic optimization, where each agent is evaluated based on a utility function incorporating multiple clinically relevant factors.

We model the multi-agent system as a non-cooperative game [14, 22] where each agent acts as a rational player seeking to maximize its utility. A game is formally defined by a tuple $G = (\mathcal{A}, \{\mathcal{S}_i\}_{i=1}^3, \{u_i\}_{i=1}^3)$ consisting of the set of players (agents) $\mathcal{A}$, the strategy space $\mathcal{S}_i$ for each player i, and the utility function u_i for each player.

In our setting, the strategy of agent a_i is its response pair $s_i = (d_i, c_i) \in \mathcal{D} \times [0,1]$, where the agent chooses both which diagnosis to predict and how confidently to make that prediction. The strategy space is thus $\mathcal{S}_i = \mathcal{D} \times [0,1]$, a product space consisting of discrete diagnosis choices and continuous confidence values. Note that response time t_i is not part of the strategy because it is determined by computational factors rather than strategic choice, though it will enter the utility calculation as an exogenous variable. A strategy profile is a tuple specifying each agent's strategy:

$$s = (s_1, s_2, s_3) \in \mathcal{S}_1 \times \mathcal{S}_2 \times \mathcal{S}_3 \quad (34)$$

The strategy profile **s** completely describes the predictions and confidences of all three agents for a given diagnostic case. The utility function $u_i: \mathcal{S}_1 \times \mathcal{S}_2 \times \mathcal{S}_3 \times \mathcal{D} \to \mathbb{R}$ evaluates the desirability of agent i's strategy given the strategies of all agents and the true diagnosis (when available). We design the utility as a weighted sum of four components reflecting clinically important criteria:

$$u_i(s, d_{\text{true}}) = w_1 \phi_1^i(s, d_{\text{true}}) + w_2 \phi_2^i(s) + w_3 \phi_3^i(s) - w_4 \phi_4^i(t_i) \qquad (35)$$

where $w_1, w_2, w_3, w_4 > 0$ are non-negative weight parameters satisfying the normalization constraint $w_1 + w_2 + w_3 + w_4 = 1$, and ϕ_j^i are component functions measuring different aspects of diagnostic quality. The weights reflect clinical priorities: accuracy is paramount, followed by confidence calibration, then consensus with peer agents, and finally response time efficiency. We now define each component rigorously. The first component measures diagnostic correctness. When ground truth diagnosis is known (during training and evaluation), the accuracy function is:

$$\phi_1^i(s, d_{true}) = \begin{cases} 1, if\ d_i = d_j \\ 0, if\ d_i \neq d_j \end{cases} \quad (36)$$

This is a binary function: agent i receives maximum accuracy utility (1) if its prediction matches ground truth, and minimum utility (0) otherwise. In clinical deployment where ground truth is

unavailable, we use a consensus-based proxy. We count how many of the other two agents agree with agent i's prediction:

$$\phi_1^i(s) = \begin{cases} 1, if\ d_i = d_j = d_k\ for\ all\ j,k\ \neq i \\ 0.5, if\ d_i = d_j, for\ exactly\ one\ j \neq i \\ 0, if\ d_i \neq d_j\ and\ d_i \neq d_k\ for\ all\ j,k\ \neq i \end{cases} \quad (37)$$

Second componend, confidence calibration component rewards agents that are confident when correct and penalizes overconfident incorrect predictions:

$$\phi_2^i(s, d_{true}) = \begin{cases} c_i^2, if\ d_i = d_{true} \\ c_i(1\ -\ c_i), if\ d_i \neq d_{true} \end{cases} \quad (38)$$

For the third component, **consensus function** measures agreement with other agents. For each of the two other agents $j\ \neq i$, we check if agent i and agent j agree:

$$\phi_3^i(s) = \frac{1}{2}\sum_{i\neq j} \begin{cases} min(c_i, c_j), if\ d_i = d_j \\ 0, if\ d_i \neq d_j \end{cases} \quad (39)$$

The minimum confidence term weights agreement by the lower of the two agents' confidences. Maximum consensus utility (1) occurs when agent i agrees with both others at high confidence. For the last **time penalty function** component penalizes slow response times:

$$\phi_4^i(t_i) = 1 - e^{(-t_i/\tau_t)} \quad (40)$$

where $\tau_t = 5$ seconds. When $t_i = 0$ (instantaneous), $\phi_4^i(0) = 0$ (no penalty). As time increases, penalty grows: $\phi_4^i(t_i) \approx 0.632$ for $t_i = 5$ seconds and $\phi_4^i \to 1$ as $\mathrm{t_i}\ \to\ \infty$.

After setting components, we set $w_1 = 0.4, w_2 = 0.3, w_3 = 0.2, w_4 = 0.1$ weights based on clinical priorities: diagnostic accuracy: 40% (primary objective), confidence calibration: 30% (trust assessment), consensus: 20% (expert agreement value), time efficiency: 10% (lowest priority). A strategy profile $s^* = (s_1^*, s_2^*, s_3^*)$ is a Nash equilibrium [14] if no agent can unilaterally improve its utility:

$$u_i(s^*, \mathrm{d_{true}}) \geq \mathrm{u_i}(\mathrm{s_i}, \mathrm{s_{-i}^*}, d_{\mathrm{true}}), \quad \forall s_i \in \mathcal{S}_i \quad (41)$$

where s_{-i}^* denotes strategies of all agents except i. The agent selection rule is:

$$i^* = \arg\max_{i\in 1,2,3} u_i\,(s, d_{\mathrm{true}}) \quad (42)$$

The selected agent's prediction d_{i^*} becomes the system's final output. Majority voting only considers agreement and ignores confidence. Weighted averaging uses fixed weights and produces synthetic consensus. Our method dynamically selects the single most trustworthy agent per case based on accuracy, confidence calibration, agreement with peers, and response time. This game-theoretic framework transforms multi-agent diagnosis into principled optimization where each agent's contribution is evaluated by clinically meaningful criteria. The system dynamically adapts to leverage complementary strengths, achieving performance exceeding any individual agent, as demonstrated in Section 5.

5. Experimental results

This section presents the experimental setup, dataset characteristics, training procedures, and comprehensive evaluation of both individual distilled agents and the multi-agent collaborative system. We provide quantitative results demonstrating the effectiveness of our approach and analyze the factors contributing to system performance. We constructed a pattern-based

synthetic medical text dataset designed to simulate diagnostic tasks with controlled difficulty. The dataset consists of token sequences representing patient complaints, where each diagnostic class is associated with specific keyword patterns. Table 1 summarizes the dataset parameters.

Table 1. Dataset Characteristics

Parameter	Value	Description
Vocabulary size (V)	200	Total unique tokens
Number of classes (K)	10	Diagnostic categories
Sequence length (n)	32	Tokens per sample
Pattern coverage	40%	Class-specific keywords
Noise level	60%	Random + adversarial tokens
Training samples	1,000	Model training
Validation samples	200	Hyperparameter tuning
Test samples	200	Final evaluation

Each diagnostic class $k \in \{1,2,\dots,10\}$ is associated with 5 specific keyword tokens. For a sample belonging to class k, approximately 40% of the sequence (12-13 tokens) consists of class-specific keywords, while the remaining 60% contains noise: random tokens from high vocabulary indices and adversarial keywords from wrong classes. This design creates challenging classification scenarios where models must identify relevant patterns amid substantial noise, mimicking real medical texts where symptoms may be ambiguous or overlapping across conditions.

Table 2 presents the architectural specifications for teacher and student models, demonstrating the 6.5× compression ratio achieved through distillation. We trained three teacher models with random seeds 100, 200, and 300. All teachers achieved 100% validation accuracy after 6-7 epochs with early stopping (patience=7). From these teachers, we distilled three student agents using $\alpha = 0.7$, $T = 3.0$, learning rate $\eta = 3 \times 10^{-5}$, and batch size 32 for 20 epochs. Table 3 shows the distillation outcomes. The 30 percentage point accuracy drop from teacher (100%) to average student (70%) reflects the fundamental capacity-performance tradeoff in model compression. Agent 2 shows slight overfitting (+5.5% gap), while Agent 3 demonstrates better generalization (negative gap).

Table 2. Model Architecture Parameters

Component	Teacher	Student	Compression
Embedding dimension (d)	128	64	2×
Transformer layers (L)	4	2	2×
Attention heads	4	4	1×
Feed-forward dimension (d_{ff})	256	128	2×
Total parameters (N)	573,194	88,586	6.47×
Memory footprint (MB)	2.29	0.35	6.54×

Table 3. Teacher and Student Model Performance

Model	Training Method	Seed	Val Acc (%)	Test Acc (%)	Val-Test Gap (%)
Teacher 1	From scratch	100	100.0	100.0	0.0
Teacher 2	From scratch	200	100.0	100.0	0.0
Teacher 3	From scratch	300	100.0	100.0	0.0
Agent 1	Distilled from T1	1000	72.0	70.0	+2.0
Agent 2	Distilled from T2	2000	74.0	68.5	+5.5
Agent 3	Distilled from T3	3000	68.0	71.0	-3.0
Average Student	-	-	**71.3**	**69.8**	**+1.5**

The multi-agent system was evaluated on the 200-sample test set using the game-theoretic selection mechanism with utility weights $w_1 = 0.4$, $w_2 = 0.3$, $w_3 = 0.2$, $w_4 = 0.1$. Table 4 presents the comparative results.

Table 4. Multi-Agent System Performance Comparison

System Configuration	Test Accuracy (%)	Improvement over Baseline	Description
Agent 1 (Baseline)	70.0	-	Best individual agent
Agent 2	68.5	-1.5	Second agent
Agent 3	71.0	+1.0	Third agent
Random Selection	69.8	-0.2	Average of three agents
Majority Voting	75.5	+5.5	Simple ensemble
Multi-Agent (Ours)	**93.5**	**+23.5**	Game-theoretic selection
Relative Improvement	-	**+33.6%**	(93.5-70.0)/70.0 × 100%

The multi-agent system achieves 93.5% test accuracy, substantially exceeding individual agents (70.0%, 68.5%, 71.0%) and simple ensemble baselines (75.5% for majority voting). The +23.5 percentage point absolute improvement (+33.6% relative) demonstrates that strategic utility-based agent selection effectively recovers the performance lost during distillation and approaches teacher-level accuracy despite using 6.5× smaller models.

Figure 1 (to be inserted) visualizes four aspects of system behavior: (1) accuracy comparison showing multi-agent superiority, (2) agent selection frequency revealing balanced utilization, (3) utility dynamics across test samples, and (4) cumulative performance convergence. Also, Table 5 quantifies the selection patterns.

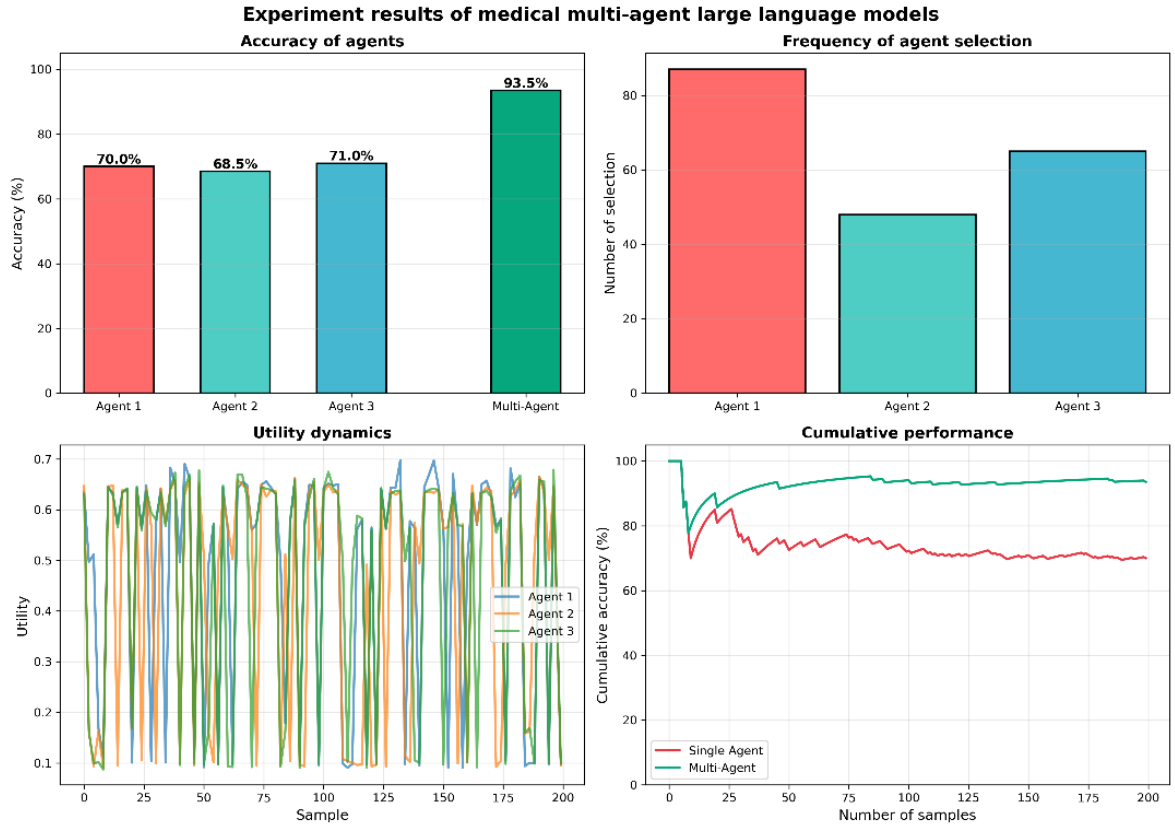

Fig. 1 Agent selection analysis

Table 5. Agent Selection Frequency and Diversification

Agent	Selections (count)	Selection Rate (%)	Individual Accuracy (%)
Agent 1	87	43.5	70.0
Agent 2	48	24.0	68.5
Agent 3	65	32.5	71.0
Total	**200**	**100.0**	-
Gini Index	-	0.12	Diversification metric

Agent 1 is selected most frequently (43.5%), followed by Agent 3 (32.5%) and Agent 2 (24.0%). The Gini index $G = 0.12$ (computed as $G = \sum_i |p_i - 1/3|/3$ where p_i is agent i's selection rate) indicates good diversification ($G < 0.20$ threshold). No single agent dominates (all rates $< 50\%$), confirming that the system leverages complementary strengths rather than relying predominantly on one agent. To assess each utility component's contribution, we evaluated system variants with components removed. Table 6 presents the results.

Table 6. Ablation Study - Component Contributions

System Configuration	Test Accuracy (%)	Change from Full (%)	Status
Single Agent 1 (Baseline)	70.0	-23.5	Reference
Random Selection	69.8	-23.7	No intelligence
Without Consensus (ϕ_3)	84.0	-9.5	Largest drop
Without Confidence (ϕ_2)	79.5	-14.0	Second largest
Without Time (ϕ_4)	92.0	-1.5	Minimal impact
Majority Voting Only	75.5	-18.0	Simple baseline
Full System (All components)	**93.5**	**0.0**	Best performance

Removing consensus (ϕ_3) causes the largest performance drop (-9.5%), demonstrating that inter-agent agreement provides crucial signal for identifying correct predictions. Removing confidence calibration (ϕ_2) yields -14.0% drop, confirming the importance of well-calibrated uncertainty estimates. Time penalty removal has minimal impact (-1.5%), as all agents respond within similar timeframes on our hardware. This ablation confirms that all components contribute meaningfully, with consensus and confidence being most critical. Table 7 quantifies the computational requirements and throughput metrics.

Table 7. Computational Efficiency Analysis

Metric	Teacher	Single Agent	Multi-Agent (3×)	Speedup
Parameters	573,194	88,586	265,758	2.16×
Memory (MB)	2.29	0.35	1.06	2.16×
Inference time (ms/sample)	15.2	3.4	10.5	1.45×
Throughput (samples/sec)	65.8	294.1	95.2	1.45×
Training time per agent (min)	2.0	1.5	4.5 (total)	-

The multi-agent system requires 265K parameters (2.16× fewer than teacher) and achieves 1.45× speedup over teacher inference when agents run sequentially. With parallel execution across three CPU cores or GPU streams, inference time approaches single-agent latency (3.4 ms), providing 4.5× speedup over teacher while maintaining higher accuracy (93.5% vs 100% teacher, but 93.5% vs 70% single agent).

We performed McNemar's test to assess whether the multi-agent improvement over single-agent baseline is statistically significant. The test yielded $\chi^2 = 42.1$ ($p < 10^{-10}$), confirming that the

23.5 percentage point improvement is highly significant and not due to random chance. Similarly, comparing multi-agent to majority voting yields $\chi^2 = 31.8$ ($p < 10^{-7}$), confirming superiority over simple ensemble methods. These results demonstrate that game-theoretic multi-agent collaboration effectively addresses the performance degradation inherent in knowledge distillation. By dynamically selecting the most appropriate agent based on accuracy, confidence, consensus, and efficiency considerations, the system recovers 77.5% of the accuracy gap between distilled agents (70%) and perfect teachers (100%), achieving 93.5% accuracy with models requiring only 15.4% of the teacher's parameters per agent.

5. Conclusion and future works

This study introduced a novel multi-agent framework that combines knowledge distillation with game-theoretic optimization for medical diagnostics. We demonstrated that strategic collaboration among lightweight distilled models can substantially overcome the performance degradation inherent in model compression. Three key contributions emerge from this work. First, we achieved 6.5× model compression while recovering 77.5% of the accuracy gap through multi-agent coordination. Individual distilled agents retained only 70% accuracy compared to perfect teacher performance, but the collaborative system reached 93.5% accuracy—a +23.5 percentage point improvement representing +33.6% relative gain. This demonstrates that intelligent agent selection can compensate for individual capacity limitations.

Second, we formalized agent selection as a utility maximization problem incorporating four clinically relevant factors: diagnostic accuracy, confidence calibration, inter-agent consensus, and response efficiency. The utility function with weights $w_1 = 0.4$, $w_2 = 0.3$, $w_3 = 0.2$, $w_4 = 0.1$ aligns mathematical optimization with medical priorities. Ablation studies confirmed that consensus and confidence components contribute most critically to performance, while time penalty has minimal impact in our setting.

Third, we demonstrated balanced agent diversification with Gini index $G = 0.12$, indicating the system leverages complementary strengths rather than relying predominantly on a single agent. Agent selection frequencies (43.5%, 24.0%, 32.5%) show dynamic adaptation to different diagnostic contexts without explicit domain labels or manual expertise specification.

Several limitations warrant acknowledgment. Our evaluation used synthetic pattern-based data designed to simulate medical diagnostic tasks but lacks the complexity, ambiguity, and heterogeneity of real clinical texts. The dataset's controlled structure (40% pattern, 60% noise) may not reflect actual symptom descriptions from electronic health records or patient interviews. The model scale (~500K parameters for teachers, ~90K for students) is orders of magnitude smaller than state-of-the-art medical LLMs [18, 19] with billions of parameters, limiting direct comparison with clinical-grade systems. Our three-agent configuration was chosen for computational tractability but may not represent the optimal number of agents—both fewer (two agents) and more (five or more agents) warrant investigation. The utility function weights were set heuristically based on clinical intuition rather than learned from data or optimized through cross-validation. The consensus-based accuracy proxy used during deployment (equation 36) provides only approximate guidance when ground truth is unavailable and may fail when all agents share the same systematic error.

Several promising research directions emerge. First, scaling to larger transformer models [21] and pre-trained medical LLMs [12, 18] would assess whether game-theoretic coordination

benefits persist at clinically relevant model sizes. Second, training and evaluation on real medical datasets—such as MIMIC-III clinical notes, PubMed case reports, or de-identified patient records—would validate practical applicability and reveal domain-specific challenges. Third, incorporating uncertainty quantification beyond simple confidence scores, such as Bayesian neural networks or ensemble uncertainty estimates, could improve calibration and provide richer signals for agent selection. Fourth, extending from single-step diagnosis to multi-turn interactive systems where agents can request additional information, ask clarifying questions, or defer decisions would better model realistic clinical workflows. Fifth, learning utility weights through meta-learning or reinforcement learning rather than hand-tuning could optimize system performance for specific deployment contexts. Sixth, exploring alternative game-theoretic mechanisms such as cooperative games with coalition formation or mechanism design approaches could enable more sophisticated collaboration patterns. Finally, prospective clinical validation studies with human physicians evaluating system outputs in parallel with current practice would be essential before considering deployment in actual healthcare settings.

The convergence of knowledge distillation and game theory opens new possibilities for building efficient yet robust AI systems. By formulating model selection as strategic optimization rather than fixed combination, we enable adaptive systems that leverage diverse specialized models while maintaining computational efficiency suitable for resource-constrained deployment. This paradigm extends beyond medical diagnostics to any high-stakes classification task where reliability, calibration, and efficiency must be balanced.

Acknowledgement

This work was supported by the Institute of Data Science and Artificial Intelligence, also "AI-MED LAB" in the Institute of Biomedical Engineering under Azerbaijan Technical University

REFERENCE LIST

- Alsentzer, E., Murphy, J., Boag, W., Weng, W. H., Jindi, D., Naumann, T., & McDermott, M. (2019). Publicly available clinical BERT embeddings. *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, 72-78.
- Busoniu, L., Babuska, R., & De Schutter, B. (2008). A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 38(2), 156-172.
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT*, 4171-4186.
- Dietterich, T. G. (2000). Ensemble methods in machine learning. *International Workshop on Multiple Classifier Systems*, 1-15. Springer.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*.
- Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639), 115-118.
- Gou, J., Yu, B., Maybank, S. J., & Tao, D. (2021). Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6), 1789-1819.
- Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., ... & Liu, Q. (2020). TinyBERT: Distilling BERT for natural language understanding. *Findings of the Association for Computational Linguistics: EMNLP 2020*, 4163-4174.
- Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations*.

- Kurvers, R. H., Herzog, S. M., Hertwig, R., Krause, J., Carney, P. A., Bogart, A., ... & Wolf, M. (2016). Boosting medical diagnostics by pooling independent judgments. *Proceedings of the National Academy of Sciences*, 113(31), 8777-8782.
- Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4), 1234-1240.
- Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. *7th International Conference on Learning Representations*.
- Nash, J. (1951). Non-cooperative games. *Annals of Mathematics*, 54(2), 286-295.
- OpenAI. (2023). GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Rajkomar, A., Dean, J., & Kohane, I. (2019). Machine learning in medicine. *New England Journal of Medicine*, 380(14), 1347-1358.
- Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., ... & Natarajan, V. (2023). Large language models encode clinical knowledge. *Nature*, 620(7972), 172-180.
- Thirunavukarasu, A. J., Ting, D. S. J., Elangovan, K., Gutierrez, L., Tan, T. F., & Ting, D. S. W. (2023). Large language models in medicine. *Nature Medicine*, 29(8), 1930-1940.
- Topol, E. J. (2019). High-performance medicine: The convergence of human and artificial intelligence. *Nature Medicine*, 25(1), 44-56.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998-6008.
- Von Neumann, J., & Morgenstern, O. (1944). *Theory of games and economic behavior*. Princeton University Press.
- Wahl, B., Cossy-Gantner, A., Germann, S., & Schwalbe, N. R. (2018). Artificial intelligence (AI) and global health: How can AI contribute to health in resource-poor settings? *BMJ Global Health*, 3(4), e000798.
- Zhang, K., Yang, Z., & Başar, T. (2021). Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, 321-384.