A Hybrid PINN-DE Framework for Data-Driven Parameter Estimation of Tumor-Immune Dynamics in Bladder Cancer

Antonio Mastroberardino[1◕,*], Adam Glick[2◕]

1 Department of Mathematical Sciences, United States Military Academy, West Point, NY, USA **2** Group G, PeopleTec, Inc., Huntsville, AL, USA

◕These authors contributed equally to this work.
* antonio.mastroberardino@westpoint.edu

Abstract

Bladder cancer presents significant clinical challenges due to its complex immune microenvironment and highly heterogeneous response to treatments. To create accurate, individualized models of disease progression, we first construct a system of Ordinary Differential Equations (ODEs) that captures tumor–immune interactions. We address the challenge of estimating unknown parameters by performing a rigorous comparative analysis of two heuristic optimization methods: Differential Evolution (DE), a robust global optimization algorithm, and Physics-Informed Neural Networks (PINN), a novel machine learning framework that embeds ODE constraints into its loss function. Our findings provide a critical evaluation of the computational efficiency and accuracy of each method for parameterizing biological ODE systems. This study validates the power of hybrid machine learning approaches in mathematical oncology, yielding a data-driven model of bladder cancer progression with direct potential for optimizing personalized treatment strategies.

Author summary

Bladder cancer remains a major global health threat, characterized by highly unpredictable responses to treatment and a high likelihood of recurrence. To better predict how a patient's disease will progress, researchers use mathematical models that simulate the "war" between cancer cells and the immune system. However, these models are only useful if they can be accurately tuned to a specific patient's data—a process called parameter estimation.

This task is notoriously difficult because clinical data is often sparse and noisy, making it hard to find the right settings for the model. In this study, we developed a novel computational framework that combines a traditional "survival of the fittest" optimization algorithm (Differential Evolution) with Physics-Informed Neural Networks (PINNs), a specialized architecture designed to embed physical constraints directly into the learning process.

By "teaching" the AI the underlying biological laws of cancer growth, our hybrid approach can accurately estimate a patient's unique disease parameters even when raw data is limited. We validated this method using a "virtual patient" system derived from real-world clinical trials. Our results show that this hybrid approach provides a more robust and reliable way to personalize cancer models, offering a powerful new tool for doctors to simulate and optimize individual treatment plans before they are even administered.

February 16, 2026

Introduction

Bladder cancer is a significant global health challenge, which ranks as the ninth most common cancer worldwide with approximately 614,000 new cases and 221,000 deaths annually [1]. Incidence and mortality are projected to rise substantially over the coming decades due to aging populations and persistent risk factors such as tobacco exposure. Projections by the World Health Organization indicate that new cases will more than double to over 1.3 million by 2050, with deaths nearly doubling to 440,000 [1]. Clinical management depends on disease stage, classified primarily as non-muscle-invasive bladder cancer (NMIBC), which accounts for approximately 75% of cases, muscle-invasive bladder cancer (MIBC), or metastatic disease [2].

For patients with NMIBC, the primary first-line treatment is transurethral resection of bladder tumor (TURBT) followed by risk-stratified adjuvant intravesical therapy [3]. Low-risk NMIBC is typically managed with a single, immediate intravesical instillation of chemotherapy with either gemcitabine or mitomycin [4–8]. For intermediate- and high-risk disease, the standard first-line treatment is a six-week induction course of intravesical Bacillus Calmette-Guérin (BCG) therapy [9–11]. For BCG-unresponsive patients, options include radical cystectomy, intravesical gemcitabine/docetaxel, or pembrolizumab for those ineligible for surgery [12, 13].

MIBC and metastatic disease require aggressive systemic interventions, including cisplatin-based chemotherapy, radical cystectomy, or increasingly, immune checkpoint inhibitors (ICIs) such as pembrolizumab and atezolizumab [14–24]. While platinum-based chemotherapy has traditionally been the first-line palliative treatment for metastatic disease, the landscape is rapidly evolving. For example, recent trials indicate that combination approaches—such as enfortumab vedotin with pembrolizumab [25, 26] or chemotherapy followed by nivolumab maintenance [27]—significantly outperform traditional chemotherapy in reducing disease progression and mortality.

Across all disease stages, treatment outcomes vary substantially from patient to patient. NMIBC exhibits recurrence rates as high as 70%, high-risk NMIBC can progress to MIBC, and metastatic disease shows wide variability in response to systemic therapies. This heterogeneity reflects complex dynamics between tumor cells, immune effector populations, and treatment-induced modulation of the microenvironment. Mechanistic mathematical models, particularly systems of ordinary differential equations (ODEs), offer a principled framework to capture these interactions and inform treatment design.

In bladder cancer, ODE models have been applied to study Bacillus Calmette–Guérin (BCG) immunotherapy, elucidating the role of immune effector activation in tumor control and informing optimal dosing schedules [28–33]. While these models offer valuable insight into treatment efficacy and optimization of dosing schedules, their predictive utility in clinical settings is often limited by uncertainty in key parameters, such as tumor growth rates, immune killing efficacy, and therapy response coefficients.

In this study, we develop an ODE-based model of tumor–immune interactions in bladder cancer and validate it using clinical datasets. Our model incorporates pharmacokinetic behavior for chemotherapy and immunotherapy using a dynamic function that more accurately represents the absorption and clearance of the drug. To overcome the challenge of parameter estimation, we present a hybrid computational approach that combines Differential Evolution (DE) [34–37]—a robust global optimization algorithm—with Physics-Informed Neural Networks (PINNs) [38–41], which embed physical laws into neural network architectures. DE excels in navigating nonconvex, high-dimensional parameter spaces without requiring smooth or differentiable objective functions. PINNs, on the other hand, integrate mechanistic knowledge with data-driven learning, enabling robust parameter estimation even under data scarcity.

We perform a comparative analysis by systematically evaluating the accuracy, robustness, and computational performance of DE and PINNs for parameter estimation. This study provides insight into the strengths and limitations of heuristic optimization versus hybrid machine-learning approaches in mathematical oncology. Our results demonstrate how data-driven parameterization can enhance the predictive capability of mechanistic tumor models and support personalized treatment planning in bladder cancer.

Model Formulation

We develop a deterministic mathematical framework to simulate the interactions between the bladder tumor burden and the host immune response. This model allows for the investigation of tumor progression under natural immune surveillance and the subsequent response to therapeutic interventions.

Tumor-Immune Dynamics Without Therapy

Let $T(t)$ and $I(t)$ denote the tumor cell population and the immune effector cell population (e.g., cytotoxic T-lymphocytes and Natural Killer cells) at time t, respectively. We propose a coupled system of ordinary differential equations (ODEs) based on the following biological assumptions:

1. **Tumor Growth:** In the absence of an immune response, tumor cells exhibit logistic growth, characterized by an intrinsic growth rate r and a carrying capacity $1/b$.

2. **Immune Influx and Homeostasis:** The immune population is maintained by a constant source rate σ (representing influx from bone marrow or lymph nodes) and decays at a natural death rate δ.

3. **Immunosurveillance:** Immune cells eliminate tumor cells upon contact. We model this interaction using Holling Type I functional response with a rate constant α_{ti}.

4. **Immune Recruitment and Inhibition:** At low tumor burden, tumor-associated antigens stimulate immune recruitment at a rate α_{it}. However, increasing tumor burden promotes an immunosuppressive microenvironment. We model this effect phenomenologically by introducing a threshold $1/\beta$ at which immune recruitment is fully suppressed by the tumor.

The inhibition of the immune system by a tumor is commonly described as immune tolerance [42], during which the immune response becomes incapable of recognizing

tumor antigens as foreign or mounting an effective attack. Tumors develop immune tolerance through multiple, overlapping mechanisms that collectively suppress or evade antitumor immunity [43–45]. A common mechanism involves altered antigen expression and presentation, whereby malignant cells downregulate tumor-associated antigens or MHC class I molecules to avoid recognition by cytotoxic T lymphocytes. In parallel, many tumors activate immune checkpoint pathways by overexpressing ligands such as PD-L1 and PD-L2, which bind to PD-1 on effector T cells and transmit inhibitory signals that attenuate proliferation and cytotoxic function.

The tumor microenvironment further reinforces tolerance through immunosuppressive signaling [46, 47]. Tumor and stromal cells secrete inhibitory cytokines such as transforming growth factor-β (TGF-β) and interleukin-10 (IL-10) to inhibit antigen presentation and effector T-cell activation. These signals also promote the recruitment and expansion of regulatory immune populations including regulatory T cells (Tregs), myeloid-derived suppressor cells (MDSCs), and M2-polarized macrophages, which collectively maintain an anti-inflammatory environment. Exhausted T cells remain present but express high levels of inhibitory receptors, preventing them from mounting effective responses. Together, these processes work synergistically to create a tolerant immune landscape in which tumors can grow and evolve unchecked.

Fig. 1 presents a schematic illustration of the key biological components included in the model and their interactions. The governing system of ODEs based on these assumptions is given by

$$\frac{dT}{dt} = rT\,(1 - bT) - \alpha_{ti} T I, \tag{1}$$

$$\frac{dI}{dt} = \sigma - \delta I + \alpha_{it} I T (1 - \beta T). \tag{2}$$

See Table 1 for a full description of the system parameters.

Parameter	**Units**	**Description**
r	week^{-1}	Intrinsic tumor growth rate
b	cells^{-1}	Tumor carrying capacity parameter ($1/b$)
α_{ti}	cells^{-1} week^{-1}	Immune-mediated tumor killing rate
σ	cells week^{-1}	Constant influx rate of immune effector cells
δ	week^{-1}	Natural death rate of immune effector cells
α_{it}	cells^{-1} week^{-1}	Tumor-induced immune recruitment rate
β	cells^{-1}	Immune inhibition (tolerance) strength
ϵ_t	week^{-1}	Maximal chemotherapy-induced tumor kill rate
η_i	week^{-1}	Maximal immunotherapy-induced immune activation rate
ω_ϵ	week	Chemotherapy absorption half-life
τ_ϵ	week	Chemotherapy clearance half-life
ω_η	week	Immunotherapy absorption half-life
τ_η	week	Immunotherapy clearance half-life
C_ϵ	–	Chemotherapy dose normalization constant
C_η	–	Immunotherapy dose normalization constant

Table 1. Model parameters and biological interpretation.

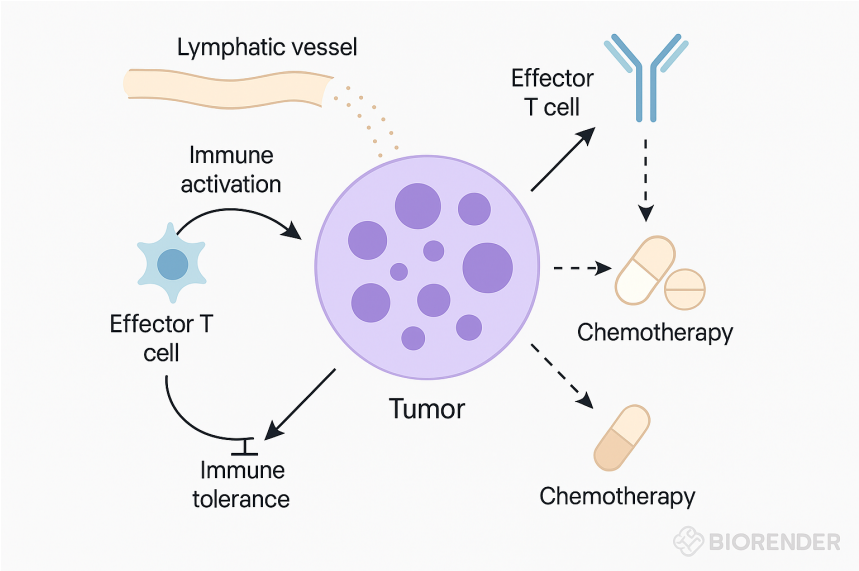


Fig 1. Schematic representation of tumor–immune interactions and therapeutic interventions. The tumor microenvironment recruits effector T cells via lymphatic signaling (immune activation), but tumor progression induces immune tolerance through inhibitory pathways. Immunotherapy (antibody-based) and chemotherapy act on tumor and immune components to restore antitumor activity and reduce tumor burden. Image created at bioRender.com.

Model with Therapeutic Intervention

We now include therapy terms in equations (1)−(2) to model a realistic administration of immunotherapy and chemotherapy. To account for the fact that clinical dosing is discrete and transient, we model the *effective drug concentration* at the tumor site using pharmacokinetic decay functions.

Let $A(t)$ and $B(t)$ represent the time-dependent efficacy of chemotherapy and immunotherapy, respectively. This formulation is mathematically equivalent to the convolution-based impulse–response model introduced in [48], but is written here in a form that explicitly highlights treatment accumulation and clearance. The extended system is defined as

$$\frac{dT}{dt} = rT(1 - bT) - \alpha_{ti}TI - A(t)T, \tag{3}$$

$$\frac{dI}{dt} = \sigma - \delta I + \alpha_{it}IT(1 - \beta T) + B(t)I, \tag{4}$$

where the chemotherapy term $-A(t)T$ represents the direct fractional kill of tumor cells, while the immunotherapy term $+B(t)I$ represents the enhanced proliferation or reduced exhaustion of effector cells.

Following the approach proposed by Glick and Mastroberardino [48], we model the drug action not as a constant, but as a dynamic function of uptake and clearance, thereby providing a more accurate representation of the drug's pharmacokinetics. For a single dose administered at time $t = 0$, we define the efficacy functions for

chemotherapy and immunotherapy, respectively, as

$$\Psi_\epsilon(t) = \frac{C_\epsilon}{1 - \tau_\epsilon/\omega_\epsilon}\left(e^{-\frac{\ln 2}{\omega_\epsilon}t} - e^{-\frac{\ln 2}{\tau_\epsilon}t}\right) . \tag{5}$$

$$\Psi_\eta(t) = \frac{C_\eta}{1 - \tau_\eta/\omega_\eta}\left(e^{-\frac{\ln 2}{\omega_\eta}t} - e^{-\frac{\ln 2}{\tau_\eta}t}\right) , \tag{6}$$

where ω_i is the effective absorption half-life, τ_i is the decay half-life, and C_i is a normalization constant regulating dose intensity for $i = \epsilon, \eta$.

For a treatment schedule consisting of N chemotherapy doses administered at times $\{q_0, q_1, \ldots, q_{N-1}\}$ and M immunotherapy doses at times $\{p_0, p_1, \ldots, p_{M-1}\}$, the total therapeutic functions $A(t)$ and $B(t)$ are constructed via the summation of shifted efficacy functions:

$$A(t) = \epsilon_t \sum_{n=0}^{N-1} \Psi_\epsilon(t - q_n) \cdot H(t - q_n) \tag{7}$$

$$B(t) = \eta_i \sum_{m=0}^{M-1} \Psi_\eta(t - p_m) \cdot H(t - p_m) \tag{8}$$

where $H(\cdot)$ is the Heaviside step function, and ϵ_t and η_i scale the maximal potency of chemotherapy and immunotherapy, respectively. This formulation ensures that biological outcomes depend on the realistic accumulation and clearance of therapeutic agents while remaining compatible with gradient-based and physics-informed parameter estimation techniques.

Computational Formulation

Robust training of PINNs requires an adequate amount of data in order to obtain meaningful results. However, public datasets for bladder cancer typically report aggregate survival statistics or discrete response rates per the RECIST criteria [49] rather than raw, time-series cellular measurements. To overcome this data scarcity, we develop a Monte Carlo simulation framework to synthesize longitudinal patient trajectories based on clinical trial meta-data reported by Song et al. [28].

We synthesize data using a custom built Monte Carlo script that randomly creates cell population data in the form of a time series sampled within the bounds of clinical trial results. The synthetic data is then used as the basis for parameter estimation using DE and PINNs to fit the data to Eqs. (3) – (4). All software used to generate the results presented in this work, including the code used to simulate the data, is publicly available at [50].

Data Synthesis Methodology

We synthesize data by aggregating clinical trials presented in Song [51] and summarized in Table 2. The published dataset contained only aggregate data and no individual patient information. Leveraging this data, we develop a Monte Carlo simulation in Python to generate longitudinal tumor cell count data for bladder cancer patients across various clinical trials. Each trial was defined by its unique identifier, drug, drug category, and empirical response rates, described as Complete Response (CR), Partial Response (PR), Stable Disease (SD), or Progressive Disease (PD). A total of 13 trial arms were simulated and patients were probabilistically assigned one of the four response outcomes using a multinomial distribution based on observed

clinical response rates. For reproducibility, a fixed random seed was applied to a random number generator.

Clinical Trial	# of Patients	Type of Therapy	Drug Name
NCT01375842	68	Immunotherapy	Atezolizumab
NCT02108652	310	Immunotherapy	Atezolizumab
NCT02951767	119	Immunotherapy	Atezolizumab
NCT02302807	466	Immunotherapy	Atezolizumab
NCT02302807	466	Chemotherapy	Chemotherapy
NCT01772004	293	Immunotherapy	Avelumab
NCT01928394	78	Immunotherapy	Nivolumab
NCT02387996	265	Immunotherapy	Nivolumab
NCT01848834	33	Immunotherapy	Pembrolizumab
NCT02256436	271	Immunotherapy	Pembrolizumab
NCT02256436	271	Chemotherapy	Chemotherapy
NCT02335424	370	Immunotherapy	Pembrolizumab
NCT02736266	50	Immunotherapy	Pembrolizumab

Table 2. Clinical trials aggregated by Song [51], showing the clinical trial number, number of enrolled patients, therapy modality, and drug used in each study. Only aggregate data were reported; no individual patient data were included in the original publication.

Tumor burden is modeled as a proxy for cell count, derived from tumor diameter. We assume tumor mass scales with volume and spherical geometry of the tumor so that the cell count $N(t)$ is proportional to the cube of the tumor diameter $D(t)$. The RECIST threshold for partial response of 30% reduction in diameter corresponds to a volumetric reduction of:

$$\frac{V_{final}}{V_{initial}} \approx (0.7)^3 \approx 0.343, \tag{9}$$

which implies a reduction in tumor cell count of approximately 65.7%.

We model the temporal dynamics of the tumor count for responders using a stochastic exponential decay model. For CR patients, the tumor cell count exhibits exponential decay that approaches zero defined by:

$$N_{\mathrm{CR}}(t) = N_0 \cdot e^{-t/\tau_{\mathrm{CR}}}, \quad \text{if } t \leq T_r, \tag{10}$$

where $N_0 = 10^6$ is the initial tumor cell count, $T_r \approx 6$ weeks is the response time, and τ_{CR} is a personalized decay constant such that

$$\tau_{\mathrm{CR}} \propto \frac{1}{\sqrt{\max(\mathrm{ORR}, 0.01)}} \tag{11}$$

where ORR is the trial's objective response rate. For PR patients, cell counts also decay exponentially but stabilize at a nonzero target, sampled from a normal distribution centered at $\mu = 3.43 \times 10^5$ with standard deviation $\sigma = 0.1\mu$:

$$N_{\mathrm{PR}}(t) = \begin{cases} N_0 \cdot e^{-t/\tau_{\mathrm{PR}}}, & \text{if } t \leq T_r, \\ \text{PR target}, & \text{otherwise}, \end{cases} \tag{12}$$

where

$$\tau_{\mathrm{PR}} = -\frac{T_r}{\ln(\text{PR target}/N_0)}. \tag{13}$$

The Python script is designed to support Monte Carlo generation of multiple synthetic datasets by seeding the NumPy random number generator with a configurable integer seed at the beginning of each simulation. This ensures both statistical variation and reproducibility. A single call to the script produces one or multiple realizations of the dataset depending on the parameters passed to it. Each seed samples new patient response types, target cell counts for PR, and randomized decay constants. This robust dataset was then used as the basis for data-driven statistical and machine learning techniques to determine the parameters in Eqs. (3) – (8) and test their results.

Parameter Estimation via Differential Evolution

We employ DE to establish a baseline for parameter estimation and provide robust initialization for the PINN. DE is a stochastic, population-based global optimization algorithm that excels in non-convex, non-differentiable landscapes common in biological inverse problems. Unlike gradient-based methods, which are prone to entrapment in local minima, DE evolves a population of candidate parameter vectors through mutation, crossover, and selection. This allows for an extensive exploration of the parameter space before refining the solution.

We identify a set of parameters $\boldsymbol{\theta}$ that minimize a composite loss function $\mathcal{L}(\boldsymbol{\theta})$, which quantifies the discrepancy between the ODE model predictions and the observed synthetic data. Optimization is carried out in an unconstrained parameter space $\boldsymbol{\phi}$, where physical parameters are recovered via exponential mapping and constrained within biologically plausible ranges. Therapy-specific scenarios are handled by setting degenerate bounds (e.g., $\epsilon_t = \tau_\epsilon = C_\epsilon = 0$ in immunotherapy-only cases), allowing a single objective formulation to flexibly adapt across treatment regimes.

DE Implementation and Objective Formulation

The DE optimizer maintains a population of $\mathbf{M}$ candidate solutions, each of dimension $\mathbf{D}$, where $\mathbf{D}$ is the number of parameters to be estimated. The population matrix, $\mathbf{P}$, can be represented as a stack of row vectors $\boldsymbol{\theta}_i^\top$, defined as

$$\mathbf{P} = \begin{bmatrix} \boldsymbol{\theta}_1^\top \\ \boldsymbol{\theta}_2^\top \\ \vdots \\ \boldsymbol{\theta}_M^\top \end{bmatrix} \in \mathbb{R}^{\mathbf{M}\times\mathbf{D}}. \tag{14}$$

Mutation in differential evolution generates a mutant vector for each candidate as

$$\mathbf{v}_i = \boldsymbol{\theta} r_1 + F \cdot (\boldsymbol{\theta} r_2 - \boldsymbol{\theta} r_3), \tag{15}$$

where $r_1, r_2, r_3 \in 1, \ldots, \mathbf{M}$ are distinct indices and $F > 0$ is the mutation factor controlling the step size.

Crossover produces a trial vector $\mathbf{u}_i$ by mixing components of the mutant vector with those of the target vector. For each component $j \in 1, \ldots, \mathbf{D}$, draw a random number

$$\rho_j \sim \mathcal{U}(0, 1). \tag{16}$$

The trial vector is then defined as

$$\mathrm{u}_{i,j} = \begin{cases} \mathrm{v}_{i,j}, & \text{if } \rho_j \le C_r \text{ or } j = j_{\text{rand}}, \\ \theta_{i,j}, & \text{otherwise}, \end{cases} \tag{17}$$

where $C_r \in [0,1]$ is the crossover probability and j_{rand} is a randomly chosen index ensuring that at least one parameter component is inherited from the mutant vector. This balance of mutation and crossover enhances both global exploration and local refinement in the parameter estimates.

Evaluations of the objective function are parallelized using Python's **multiprocessing** module, allowing multiple candidate solutions to be evaluated simultaneously. A shared-memory array tracks the current best solution, and a lock ensures thread-safe updates, which triggers the callback function for early stopping when appropriate. Callback functions allow early termination once a target loss or computational budget is reached, and a fixed random seed ensures reproducibility of the differential evolution runs.

For a given candidate $\boldsymbol{\theta}$, the scaled tumor–immune system is integrated numerically using **scipy.integrate.solve_ivp** with the LSODA method, which adaptively switches between stiff and non-stiff solvers. To prevent differential evolution from diverging due to excessively large tumor and immune populations as well as preventing overflow errors in our computations, the tumor and immune populations are normalized by scale factors

$$j_{\text{scale}} = 10^{\lfloor \log_{10}(\max\{j\}) \rfloor}\,, \tag{18}$$

where $j \in \{T, I\}$ so that

$$T^*(t) = \frac{T(t)}{T_{\text{scale}}}\,, \tag{19}$$

$$I^*(t) = \frac{I(t)}{I_{\text{scale}}}\,, \tag{20}$$

with initial conditions set from the first observed tumor count and mean immune count. After integration, predictions are rescaled to get the model's full result as

$$\widehat{T}(t) = T_{\text{pred}}(t) \times T_{\text{scale}}\,, \tag{21}$$

$$\widehat{I}(t) = I_{\text{pred}}(t) \times I_{\text{scale}}\,. \tag{22}$$

To account for inter-patient variability, the loss is computed as an average across all patients ($\mathbf{N}$) and evaluation times ($\mathbf{M}$) taking inspiration from [52], [53], and [54]:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) \;=\; & \lambda_{\text{lin}} \cdot \frac{1}{\mathbf{N}\cdot\mathbf{M}} \sum_{i=1}^{\mathbf{N}} \sum_{k=1}^{\mathbf{M}} \Big(\widehat{T}(t_{i,k};\boldsymbol{\theta}) - T_{\text{obs}}(t_{i,k}) \Big)^2 \\ & + \;\lambda_{\text{log}} \cdot \frac{1}{\mathbf{N}\cdot\mathbf{M}} \sum_{i=1}^{\mathbf{N}} \sum_{k=1}^{\mathbf{M}} \Big(\log\big(\widehat{T}(t_{i,k};\boldsymbol{\theta}) + \varepsilon\big) - \log\big(T_{\text{obs}}(t_{i,k}) + \varepsilon\big) \Big)^2,\end{aligned} \tag{23}$$

Here, the linear-space term emphasizes absolute deviations at larger tumor burdens, while the log-space term emphasizes relative deviations during early growth or regression phases. The small constant $\varepsilon > 0$ prevents divergence as tumor sizes approach zero. For this work, we do an average of the two loss terms by setting $\lambda_{\text{lin}} = \lambda_{\text{log}} = 0.5$ with a constant value $\varepsilon = 10^{-2}$.

Finally, if the ODE solver fails due to stiffness, instability, or unbounded growth, or if parameters fall outside admissible ranges, the objective function is assigned a large penalty (e.g., 10^6), thereby steering the optimizer away from infeasible regions of parameter space.

Parameter Constraints

To enforce positivity and stabilize the optimization, the optimizer operates on unconstrained parameters $\boldsymbol{\phi}$, with the physical parameters recovered inside the loss function via

$$\theta_i = \exp(\phi_i), \tag{24}$$
$$\theta_i \in \left[\theta_i^{\min},\ \theta_i^{\max}\right]. \tag{25}$$

This enforces positivity and constrains the estimated parameters within user-specified bounds.

The optional upper and lower bounds are passed to the DE optimizer via the **bounds** argument for additional safety. These bounds are also enforced inside the Python objective function via clamping, providing a second layer of protection against invalid parameters. The bounds chosen are specific to each parameter and typically spanning several orders of magnitude to allow exploration of wide parameter regions and identification of optimal values.

Parameter Refinement with PINN

Following coarse parameter estimation via DE, we refine the model using PINN. The DE procedure produces an optimized parameter set $\boldsymbol{\theta}^*$ that serves as the initialization for PINN. This global search reduces the risk of poor local minima and improves the likelihood of biologically meaningful convergence during PINN training. The PINN simultaneously fits tumor-immune dynamics to the data while enforcing the underlying ODE structure, enabling accurate interpolation and biologically consistent extrapolation.

Model Architecture and Loss Formulation

The data generated in Sect. is randomly separated into training and testing data using a random permutation of indices across the dataset. First, a seed is set for permutation reproducibility so that the result of the PINN's model and parameter estimations can be assessed during hyperparameter tuning. The training set is passed into the training loop and the testing set is used to assess the PINN's performance at estimating data evolution in time. For numerical stability, observed tumor and immune cell populations are normalized by their maximum observed values before training. This prevents gradients from exploding to positive and negative infinity, allowing training to commence without encountering overflow errors.

Let $\widehat{T}(t;\boldsymbol{\theta})$ and $\widehat{I}(t;\boldsymbol{\theta})$ denote the neural estimates of tumor and immune cell populations over time using PINN. The network is parameterized by trainable weights and biases (collectively denoted by $\boldsymbol{\theta}$) as well as system parameters appearing in the governing ODEs. The PINN receives as input a time point t and outputs estimates of both T and I. To enforce the ODE constraints, we require that

$$\frac{d\widehat{T}}{dt} \approx f_T(\widehat{T},\widehat{I},t;\boldsymbol{\theta}), \quad \frac{d\widehat{I}}{dt} \approx f_I(\widehat{T},\widehat{I},t;\boldsymbol{\theta}),$$

where f_T and f_I are the right-hand sides of Eqs. (3) – (4).

The network mapping is defined as

$$\begin{aligned}\left(\widehat{T}(t;\boldsymbol{\theta}),\,\widehat{I}(t;\boldsymbol{\theta})\right) &= \text{Softplus} \circ \mathcal{A}^{(\text{out})} \\ &\circ \left(\bigcirc_{\ell=1}^{L}\ \mathcal{D}_{p_\ell} \circ \text{LN}_{\gamma^{(\ell)},\beta^{(\ell)}} \circ \text{SiLU} \circ \mathcal{A}^{(\ell)} \right)(t),\end{aligned} \tag{26}$$

with nodes $n_0 = 1$, $n_\ell = 200$ for $\ell = 1, \ldots, L$, $n_{L+1} = 2$, $L = 6$ hidden layers, and $\bigcirc_{\ell=1}^{L}$ represents function composition in sequence defined by

$$\bigcirc_{\ell=1}^{L} f_\ell = f_L \circ f_{L-1} \circ \cdots \circ f_1 \,. \tag{27}$$

Each hidden layer consists of four operations applied sequentially. First, an *affine transformation* is applied,

$$\mathcal{A}^{(\ell)}(x) = W^{(\ell)} x + b^{(\ell)} \,, \tag{28}$$

$$\mathcal{A}^{(\mathrm{out})}(x) = W^{(\mathrm{out})} x + b^{(\mathrm{out})}, \tag{29}$$

with weight matrices $W^{(\ell)}$ and biases $b^{(\ell)}$ as trainable parameters. These are followed by the *SiLU activation* (also known as Swish-1),

$$\mathrm{SiLU}(x) = x\,\sigma(x) \,, \tag{30}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \tag{31}$$

which introduces smooth nonlinearity. To improve stability during training, a *layer normalization* step rescales and shifts the activations,

$$\mathrm{LN}_{\gamma,\beta}(x) = \gamma \odot \frac{x - \mu(x)}{\sqrt{\mathrm{Var}(x) + \varepsilon}} + \beta, \tag{32}$$

where $\gamma, \beta \in \mathbb{R}^{200}$ are trainable parameters, $\varepsilon > 0$ is a small constant, and μ, Var denote the feature mean and variance. $\odot$ is the Hadamard multiplication operator where if $A, B \in \mathbb{R}^{m \times n}$,

$$(A \odot B)_{ij} = A_{ij}\, B_{ij}, \quad i = 1, \ldots, m,\ j = 1, \ldots, n \,. \tag{33}$$

Finally, *dropout* is applied for regularization:

$$\mathcal{D}_{p_\ell}(x) = \frac{1}{1 - p_\ell}\, \xi^{(\ell)} \odot x \tag{34}$$

$$\xi_i^{(\ell)} \sim \mathrm{Bernoulli}(1 - p_\ell). \tag{35}$$

At inference time, dropout reduces to the identity unless Monte Carlo dropout is explicitly enabled.

At the network output, an affine transformation produces a two-dimensional vector $z = (z_1, z_2)$, which is then passed through the *Softplus* function to enforce positivity,

$$\mathrm{Softplus}(z) = \log\!\big(1 + e^{z}\big) \,, \tag{36}$$

$$\widehat{T} = \mathrm{Softplus}(z_1) \,, \tag{37}$$

$$\widehat{I} = \mathrm{Softplus}(z_2) \,. \tag{38}$$

Collecting all trainable weights and biases, the parameter set is

$$\boldsymbol{\theta} = \Big\{ W^{(\ell)}, b^{(\ell)}, \gamma^{(\ell)}, \beta^{(\ell)} \Big\}_{\ell=1}^{L} \cup \Big\{ W^{(\mathrm{out})}, b^{(\mathrm{out})} \Big\}. \tag{39}$$

For a vector of times $t \in \mathbb{R}^{T \times 1}$, the mapping acts row-wise, yielding $\big(\widehat{T}(t;\boldsymbol{\theta}), \widehat{I}(t;\boldsymbol{\theta})\big) \in \mathbb{R}^{T \times 2}$.

The total loss function is defined by

$$\mathcal{L}_{\mathrm{total}} = \lambda_{\mathrm{data}} \cdot \mathcal{L}_{\mathrm{data}} + \lambda_{\mathrm{phys}} \cdot \mathcal{L}_{\mathrm{phys}} + \lambda_{\mathrm{init}} \cdot \mathcal{L}_{\mathrm{init}} \tag{40}$$

with components

$$\begin{aligned}
\mathcal{L}_{\text{data}} &= \text{MSE}(\widehat{T}(t_d), T_{\text{obs}}) \\
\mathcal{L}_{\text{phys}} &= \text{MSE}\left(\frac{d\widehat{T}}{dt} - f_T(\widehat{T}, \widehat{I}, t)\right) + \text{MSE}\left(\frac{d\widehat{I}}{dt} - f_I(\widehat{T}, \widehat{I}, t)\right) \\
\mathcal{L}_{\text{init}} &= (\widehat{T}(0) - T_0)^2 + (\widehat{I}(0) - I_0)^2
\end{aligned}$$

where t_d are a randomly sampled subset of the observed data times. $\mathcal{L}_{\text{data}}$ is calculated for mini-batches of patients, and T_0, I_0 are the known initial conditions. The physics loss is computed at randomly sampled subset of a fine-grained time discretizations covering $t_0, \ldots, t_f$, often chosen adaptively to focus on high-residual regions.

We start with λ_j values defined in Table 3. The annealing schedule for the physics loss weight, λ_{phys}, is defined as follows, where e is the current epoch, e_{start} is the ramp-up start epoch, e_{end} is the ramp-up end epoch, $\lambda_{\text{phys, initial}}$ is the initial physics loss weight, and $\lambda_{\text{phys, final}}$ is the final physics loss weight:

$$\lambda_{\text{phys, annealed}} = 10^{\log_{10}\left(\lambda_{\text{phys, initial}}\right) + p\left(\log_{10}\left(\lambda_{\text{phys, final}}\right) - \log_{10}\left(\lambda_{\text{phys, initial}}\right)\right)}, \quad (41)$$

$$\lambda_{\text{phys}} = \begin{cases} \lambda_{\text{phys, initial}} & \text{if } e < e_{\text{start}}, \\ \lambda_{\text{phys, annealed}} & \text{if } e_{\text{start}} \leq e < e_{\text{end}}, \\ \lambda_{\text{phys, final}} & \text{if } e \geq e_{\text{end}}, \end{cases} \quad (42)$$

where the progress, p, is defined as:

$$p = \frac{e - e_{\text{start}}}{e_{\text{end}} - e_{\text{start}}}. \quad (43)$$

Parameter	**Value**	**Description**
λ_{data}	10.0	Weight for the data fidelity loss term
$\lambda_{\text{phys, initial}}$	0.01	Initial weight for the physics loss term
$\lambda_{\text{phys, final}}$	5.0	Final weight for the physics loss term
λ_{init}	0.1	Weight for the initial condition loss
e_{start}	0.02 × max epochs	Start of annealing schedule
e_{end}	0.3 × max epochs	End of annealing schedule

Table 3. Lambda weight and annealing schedule values for PINN loss terms.

Unlike a conventional PINN that treats system parameters as known constants, we treat $\boldsymbol{\theta}$ as trainable variables. Parameters $\boldsymbol{\theta}$ are initialized using the optimized values from the DE procedure, ensuring that training starts in a biologically plausible basin.

To ensure positivity and boundedness of parameters during training, we reparameterize unconstrained network weights through smooth transformations. Common choices include the exponential mapping, which enforces strict positivity, and the logistic sigmoid mapping, which constrains values to a fixed interval. In this work, we adopt the logistic sigmoid because it offers several advantages: it smoothly maps from the entire real line to a bounded interval, preventing runaway parameter growth; its derivatives are well-behaved and do not explode as in the exponential case; and it allows for explicit control of parameter ranges via $(\theta_{\min,i}, \theta_{\max,i})$. This bounded,

differentiable mapping improves numerical stability and ensures that gradient-based optimization proceeds more reliably. The sigmoid reparameterization is defined as

$$\theta_i = \theta_{\min,i} + (\theta_{\max,i} - \theta_{\min,i}) \cdot \sigma(w_i)\,, \tag{44}$$

where w_i is the unconstrained network parameter and σ is the logistic sigmoid.

Numerical Implementation and Optimization

Model optimization is performed via automatic differentiation (e.g., PyTorch **autograd**), allowing for the simultaneous update of neural network weights and ODE parameters. Training utilizes the Adam optimizer, employing a bifurcated learning rate strategy where weight decay is applied exclusively to the network weights to prevent overfitting, while ODE parameters are left unregularized to ensure unbiased physical estimation. To maintain a C^∞ differentiable loss landscape and prevent gradient vanishing, discontinuous dose-response functions are replaced with smooth approximations, such as sigmoid-smoothed Heaviside functions. We eschew piecewise linear activations like ReLU—which possess vanishing second derivatives—in favor of smooth transcendental functions. Given the high cardinality of the training dataset and the fine discretization of the temporal vector, we utilize a mini-batch training regime. To ensure global convergence across the entire temporal domain, a stochastic collocation strategy is employed, where a new subset of residual points is resampled every N epochs. This approach mitigates the stability issues often associated with mini-batching in PINNs while ensuring that the network adequately captures the transient dynamics of the tumor–immune interactions.

Glick [55] proposed a stopping condition for Maximum Likelihood Expectation Maximization (MLEM) that determines the maximum likelihood of the estimation by locating the inflection point of the divergence of the estimation between consecutive iterations. Eq. (47) shows this stopping metric, which can also be used for early termination in PINN training, where $\widehat{\mathbb{T}}^{(k+1)}$ is a vector representing the model's estimation of the data for all time points on the current epoch with the current parameter estimations, $\boldsymbol{\theta}^{(k+1)}$, and $\widehat{\mathbb{T}}^{(k)}$ is a vector representing the model's estimation of the data for all time points on the previous epoch with the previous parameter estimations, $\boldsymbol{\theta}^{(k)}$.

$$\widehat{\mathbb{T}}^{(k+1)} = \begin{pmatrix} \widehat{T}^{(k+1)}(t_0;\boldsymbol{\theta}^{(k+1)}) - \overline{\widehat{T}^{(k+1)}(t;\boldsymbol{\theta}^{(k+1)})} \\ \vdots \\ \widehat{T}^{(k+1)}(t_f;\boldsymbol{\theta}^{(k+1)}) - \overline{\widehat{T}^{(k+1)}(t;\boldsymbol{\theta}^{(k+1)})} \end{pmatrix} \tag{45}$$

$$\widehat{\mathbb{T}}^{(k)} = \begin{pmatrix} \widehat{T}^{(k)}(t_0;\boldsymbol{\theta}^{(k)}) - \overline{\widehat{T}^{(k)}(t;\boldsymbol{\theta}^{(k)})} \\ \vdots \\ \widehat{T}^{(k)}(t_f;\boldsymbol{\theta}^{(k)}) - \overline{\widehat{T}^{(k)}(t;\boldsymbol{\theta}^{(k)})} \end{pmatrix} \tag{46}$$

$$\xi^{(k+1)} = 1 - \frac{\widehat{\mathbb{T}}^{(k+1)}\widehat{\mathbb{T}}^{(k)}}{|\widehat{\mathbb{T}}^{(k+1)}|_2^2} \tag{47}$$

$\xi^{(k+1)}$ is a akin to a decorrelation measure between consecutive predictions in the tumor count. Unlike in [55] where MLEM was optimized, $\widehat{\mathbb{T}}^{(k+1)}$ is an estimation from PINN and, therefore, is not a continuously decreasing function with increasing iteration. Eq. (48) is therefore not a valid stopping condition for PINN training in this work. We have constructed a new stopping condition to define the optimal performance of the neural network, which occurs when Eq. (49) is satisfied. To ensure

that we do not prematurely break the training loop early, the network training is only halted if both Eq. (49) and Eq. (50) are satisfied for a sufficient number of epochs.

$$\frac{d^2\xi}{dk^2} = 0 \tag{48}$$

$$\frac{d\xi}{dk} \leq 0 \tag{49}$$

$$\frac{d^2\xi}{dk^2} > 0 \tag{50}$$

Requiring Eq. (50) to be strictly positive is a necessary condition for early stopping due to the momentum of the decorrelation measure. If the curvature of $\xi^{(k+1)}$ becomes negative, then the decorrelation is losing momentum and the neural network is heading toward instability between epochs rather than a plateau in its training. Eqs. (49) – (50) can be combined to create an indicator function that has the form

$$1_{\text{stop}}^{(k+1)} = \mathbf{1}\Big(|\xi^{(k+1)}| < \varepsilon_{\text{thresh}}\Big) \times \mathbf{1}\left(\frac{d\xi}{dk} \leq 0\right) \times \mathbf{1}\left(\frac{d^2\xi}{dk^2} > 0\right), \tag{51}$$

$$\textbf{Stop if} \quad \sum_{j=k-p+1}^{k} 1_{\text{stop}}^{(j)} = p\,, \tag{52}$$

where $\varepsilon_{\text{thresh}}$ is a threshold value to ensure that the decorrelation between estimations is sufficiently small and early stopping is triggered if Eq. (51) holds for p consecutive epochs.

The threshold $\varepsilon_{\text{thresh}}$ is determined rigorously by examining how the PINN prediction decorrelated from the DE prediction with increasing epoch using Eq. (47), where $\widehat{\mathbb{T}}^{(k+1)}$ is the tumor count estimated by the PINN at epoch $k+1$ and $\widehat{\mathbb{T}}^{(k)}$ is the optimized DE estimate at epoch k. Let $\{\xi_i\}_{i=1}^N$ be the decorrelation metrics across all N epochs. Then the confidence threshold $\varepsilon_{\text{thresh}}$, corresponding to a confidence level $1-\alpha$ (we take $\alpha = 0.01$ for 99% confidence), is given by the quantile (inverse cumulative distribution function) of the distribution of $|\xi|$:

$$\varepsilon_{\text{thresh}} = Q_{1-\alpha}\big(\{|\xi_i|\}_{i=1}^N\big)\,, \tag{53}$$

$$Q_{1-\alpha} = \inf\left\{x : \frac{1}{N}\sum_{i=1}^{N} 1_{|\xi_i|\leq x} \geq 1-\alpha\right\}. \tag{54}$$

Here, $1_{\{\cdot\}}$ denotes the indicator function. For the data used in this work, this procedure yields

$$\varepsilon_{\text{thresh}} \approx 0.176\,.$$

To determine p, we assume independence per epoch. We let the event that the stopping condition holds in a single epoch be defined in Eq. (51). If the stochastic fluctuations in training are approximately independent between consecutive epochs, then the probability of a false stop decreases multiplicatively with the number of consecutive epochs required. Therefore the probability of triggered early stopping spuriously over p consecutive epochs is

$$\mathbf{Pr}(\text{false stop}) = q^p\,. \tag{55}$$

If we want the probability of a spurious stop to be $\alpha_{\text{false}} \leq 1\%$, then we get an equation for patience, p, that is

$$p \geq \frac{\log(\alpha_{\text{false}})}{\log(q)}. \tag{56}$$

The parameter q represents the probability of a "spurious stop" before the network has reached a true plateau. To ensure a robust estimation, we define q empirically based on the variance observed in the Differential Evolution (DE) optimization phase:

$$q = \mathbf{Pr}(1_{\text{stop}}^{(k)} = 1 \mid \text{network has not yet plateaued})\,. \tag{57}$$

By characterizing the stochastic fluctuations inherent in the DE-derived parameter space, we obtained a value of $q = 0.240$. Substituting this into Eq. (56) yields $p \approx 3$. This selection of patience ensures that the stopping condition is grounded in the physical variability of the system rather than arbitrary heuristics.

Uncertainty Quantification via Monte Carlo Dropout

The trained PINN yields both a refined set of parameters $\boldsymbol{\theta}^\star$ and continuous, differentiable estimates of the system states over time, $\widehat{T}(t;\boldsymbol{\theta}^\star)$ and $\widehat{I}(t;\boldsymbol{\theta}^\star)$. These estimated trajectories provide a smooth functional representation of the underlying dynamics and can be used in an independent numerical integrator (e.g., SciPy's ODE solver) to verify that the inferred parameters reproduce the observed system behavior, denoted as $\boldsymbol{\theta}^\star \to T(t)$ and $I(t)$. This hybrid approach preserves the interpretability of mechanistic models while leveraging the expressive capacity of neural networks, enabling accurate reconstructions under sparse or noisy observations.

To assess model generalization and estimate predictive uncertainty, Monte Carlo dropout was employed during evaluation. Specifically, dropout layers were explicitly reactivated at test time, and the trained network was sampled multiple times ($S = 10^3$ forward passes) over a uniform test grid. Each forward pass corresponded to a stochastic realization $\widehat{T}^{(s)}(t)$ arising from a different random subset of the learned network weights being active. The ensemble of predictions $\widehat{T}^{(s)}(t)_{s=1}^{S}$ was then used to compute both the mean trajectory $\mathbb{E}[\widehat{T}(t)]$ and the associated pointwise variance $\mathrm{Var}[\widehat{T}(t)]$, providing an empirical measure of epistemic uncertainty.

This Monte Carlo dropout procedure introduces a Bayesian interpretation to the PINN's deterministic inference, allowing for statistically valid uncertainty quantification without requiring full variational Bayesian training. The resulting predictive distribution reflects both the robustness of the learned dynamics and the confidence of the model in regions of the state space that were weakly constrained by data.

To quantify the width of the pointwise confidence intervals, we computed the normalized band width (NBW), defined as the ratio of the interval width to the range of the mean trajectory:

$$\mathrm{NBW} = \frac{\overline{\mathrm{hi}(t) - \mathrm{lo}(t)}}{\max(\mathbb{E}[\widehat{T}(t)]) - \min(\mathbb{E}[\widehat{T}(t)])}, \tag{58}$$

where $\mathrm{hi}(t)$ and $\mathrm{lo}(t)$ denote the upper and lower bounds of the 95% pointwise confidence interval at each time point, and the overline indicates averaging over all time points. NBW thus provides a compact, interpretable metric to assess the typical relative width of the Monte Carlo-derived predictive uncertainty.

Results

Clinical trials taken from [51] were used to synthesize cell data for the complete response and partial response cohorts as previously described. The synthetic cell data was then used to perform parameter evaluation via differential evolution and PINN.

The two methods are compared against the raw data for accuracy of the model's prediction of the parameters. The two metrics used to determine a model's predictive accuracy were R^2, defined in Eq. (59), and mean squared error (MSE), defined in Eq. (60), where y_i is the observed data, $\widehat{y_i}$ is the model's prediction of the data, and $\overline{y_i}$ is the mean of the data.

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (y_i - \widehat{y_i})^2}{\sum_{i=1}^{N} (y_i - \overline{y_i})^2} \tag{59}$$

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \widehat{y_i})^2 \tag{60}$$

Complete Response Cohort

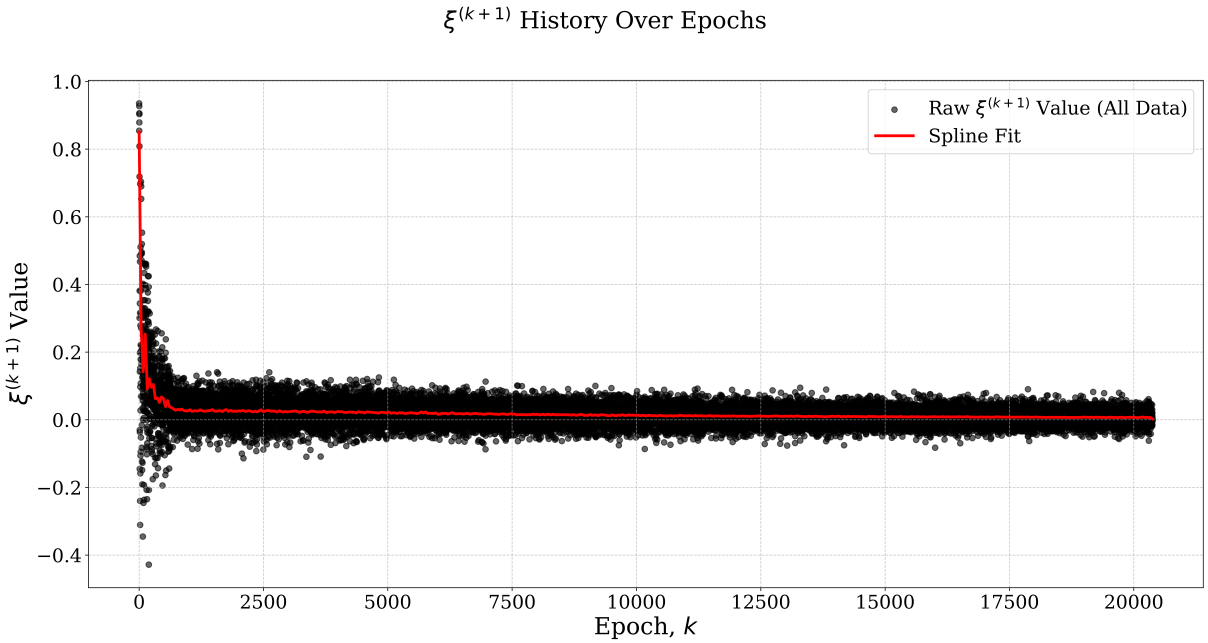


Fig 2. $\xi^{(k+1)}$ stopping condition equation raw values plotted alongside a spline interpolation of the data downsampled the iterations done during training shows high divergence in the tumor estimation between epochs that reaches stability as the number of epochs increases.

Fig. 2 shows the evolution of the conversion metric, $\xi^{(k+1)}$, plotted alongside a rolling average window taken over approximately 1% of the iterations taken during training and training was ceased after 20,378 epochs, when Eq. (52) was satisfied. $\xi^{(k+1)}$ shows high divergence between estimations in the early training history, with highly oscillatory behavior in the rolling average window, but as the epochs move forward in time $\xi^{(k+1)}$ stabilizes and the rolling average window flattens. Parameter estimates for differential evolution and PINNs are provided in Table 4, with the associated performance metrics (R^2, MSE) summarized in Table 5. Both differential evolution and PINN come to similar optimizations for parameter values in most cases for both treatment categories.

Patients treated with immunotherapy who experienced remission of their cancer and their corresponding differential evolution and PINN models can be seen in Fig. 3a. The differential evolution fits the synthesized data very closely with an R^2 value of

Parameter	Immunotherapy		Chemotherapy	
	PINN	**DE**	**PINN**	**DE**
r_t	0.75012	0.75504	0.99686	0.99734
b_t	1.47×10^{-10}	1.45×10^{-10}	4.57×10^{-10}	3.08×10^{-10}
α_{ti}	7.16×10^{-7}	7.16×10^{-7}	7.75×10^{-7}	7.75×10^{-7}
ϵ_t	—	—	0.0098	0.0084
τ_ϵ	—	—	0.04843	0.04843
ω_ϵ	—	—	2.81216	2.81216
η_i	0.0338	0.0344	—	—
τ_η	0.04490	0.04475	—	—
ω_η	5.40559	5.40561	—	—
σ_i	4587.2	4680	2398.3	2170
α_{it}	4.45×10^{-12}	4.47×10^{-12}	1.01×10^{-8}	6.21×10^{-9}
β_{it}	2.06×10^{-8}	2.06×10^{-8}	3.38×10^{-12}	3.39×10^{-12}
δ_i	0.00012	0.00010	0.00700	0.00740
C_ϵ	—	—	0.13166	0.13166
C_η	0.19884	0.20014	—	—

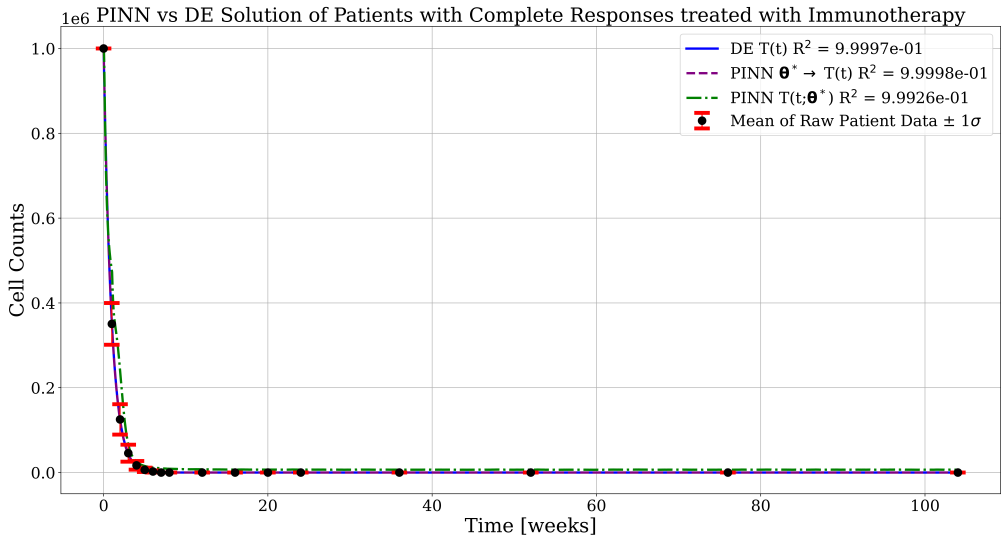
Table 4. Comparison of estimated parameters used in Eqs. (3) – (8) from PINN and Differential Evolution for patients who received a curable outcome undergoing treatment with immunotherapy or chemotherapy.

Metric	Immunotherapy			Chemotherapy		
	PINN $\widehat{T}(t;\boldsymbol{\theta}^*)$	**PINN** $\boldsymbol{\theta}^* \to T(t)$	**DE**	**PINN** $\widehat{T}(t;\boldsymbol{\theta}^*)$	**PINN** $\boldsymbol{\theta}^* \to T(t)$	**DE**
R^2	9.9921×10^{-1}	9.9998×10^{-1}	9.9997×10^{-1}	9.9980×10^{-1}	9.9997×10^{-1}	9.9997×10^{-1}
MSE	4.6573×10^{-5}	1.0899×10^{-6}	1.5546×10^{-6}	1.1750×10^{-5}	1.7385×10^{-6}	1.7509×10^{-6}

Table 5. Comparison of R^2 and MSE values for PINN and DE methods on immunotherapy and chemotherapy datasets for patients that experienced remission with therapy.

9.9997×10^{-1} and has an MSE of 1.5546×10^{-6}. PINN improves on these parameter values and when the PINN parameter estimations are used in the SciPy ODE solver, a higher R^2 value of 9.9998×10^{-1} and a lower MSE of 1.0899×10^{-6} are achieved. However, PINN's model for solving the ODE and encoding the parameters on the model has slightly worse performance as the R^2 value is lower at 9.9921×10^{-1} and the MSE is higher at 4.6573×10^{-5}. The 95% pointwise confidence intervals were extremely narrow (average NBW = 2.4%, maximum NBW = 22.9%), resulting in low empirical coverage of 29.4%. This indicates that while the model provides highly precise predictions, it underestimates predictive uncertainty in this dataset.

Patients treated with chemotherapy who experienced remission of their cancer and their corresponding differential evolution and PINN models can be seen in Fig. 3b. The differential evolution fits the synthesized data very closely with an R^2 value of 9.9998×10^{-1} and has an MSE of 9.6845×10^{-7}. PINN iterates on these parameter values and when the new estimations are used in the SciPy ODE solver, an equal R^2 value of 9.9998×10^{-1} and a statistically similar MSE of 9.6834×10^{-7} are achieved. However, like in the immunotherapy treatment cohort, PINN's model for solving the



(a) Comparison of the differential evolution results using SciPy, PINN parameter estimations with SciPy's ODE solver, and PINN parameter estimations with PINN's ODE solution plotted alongside a test dataset of patient tumor cell counts treated with immunotherapy in the complete response category.

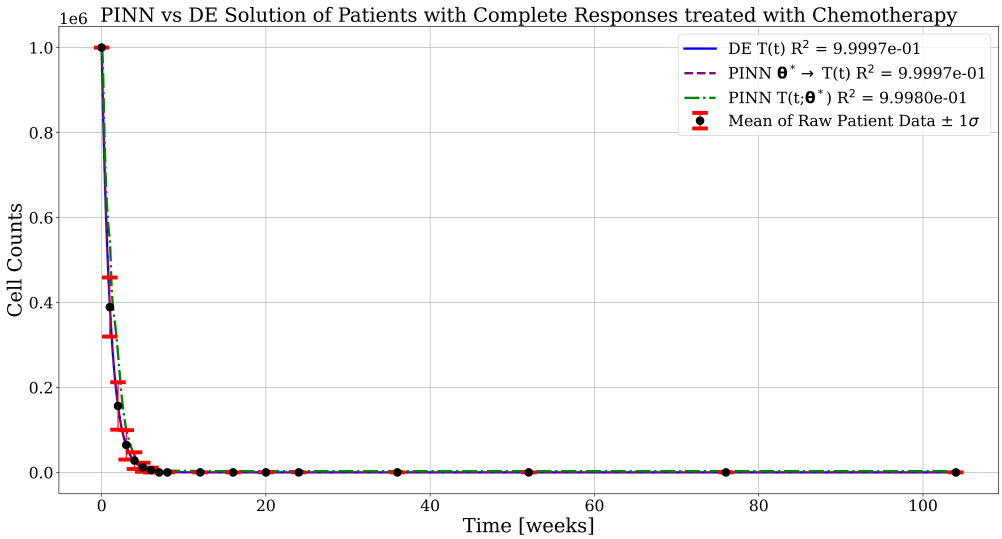
ODE has slightly worse performance as the R^2 value is lower at 9.9980×10^{-1} and the MSE is higher at 1.1750×10^{-5}. Similar to the immunotherapy cohort, the 95% pointwise confidence intervals for the chemotherapy cohort were extremely narrow (average NBW = 1.2%, maximum NBW = 27.2%), resulting in low empirical coverage of 41.2%. This indicates that while the model provides highly precise predictions, it underestimates predictive uncertainty in this dataset.

In both treatment cohorts under the complete response scenario, the PINN achieves parameter estimates that are comparable to, and in some cases better than, those obtained with differential evolution on the synthetic datasets (Fig. 3). Nonetheless, when evaluating the model trajectories, the PINN's neural solution exhibits slightly reduced accuracy relative to direct numerical integration with the SciPy ODE solver. This degradation is modest—the PINN still yields high R^2 values—but becomes apparent when comparing relative R^2 MSE across methods, where the SciPy solver consistently achieves lower error.

The modest discrepancy between the PINN and the direct SciPy ODE solution arises from the different approximation strategies employed. Whereas SciPy integrates the governing equations directly with adaptive numerical solvers that minimize local truncation error, the PINN learns an approximate mapping $t \mapsto \left(\widehat{T}(t), \widehat{I}(t)\right)$ through a neural network. This introduces additional approximation error due to finite network capacity, optimization dynamics, and the competing balance between data-fitting and physics-based residual losses. As a result, even if the parameter estimates are accurate, the neural solution may not reproduce trajectories with the same precision as direct numerical integration, particularly in low-variance or noise-free regimes where solver accuracy is very high.

Partial Response Cohort

The estimation of the parameters from differential evolution and PINN for patients who experienced a partial response for both types of treatments are listed in Table 6,



(b) Comparison of the differential evolution results using SciPy, PINN parameter estimations with SciPy's ODE solver, and PINN parameter estimations with PINN's ODE solution plotted alongside a test dataset of patient tumor cell counts treated with chemotherapy in the complete response category.

Fig 3. Comparison of differential evolution and PINN parameter and model estimations for tumor cell counts in complete response categories for immunotherapy and chemotherapy treatments.

subdivided into treatment categories, with corresponding performance metrics presented in Table 7. Both differential evolution and PINN made similar estimations for parameter values in most cases for both treatment categories.

Tumor cell counts for patients treated with immunotherapy that experienced partial remission of their cancer are plotted along with differential evolution and PINN models in Fig. 4a. The differential evolution results in a relatively high R^2 value of 9.5195×10^{-1} and an MSE of 1.8624×10^{-3}. The parameter estimations from differential evolution are used as a starting point for PINN, which then encodes the parameters on its model as it tries to fit the data by meeting the constraints of the hyperparameters λ_{data}, λ_{phys}, and λ_{init} using an annealing scheduling for λ_{phys} defined in Eq. (42). Using the PINN's parameter estimation with the SciPy's ODE solver results in a higher R^2 value of 9.5307×10^{-1} and a lower MSE of 1.8188×10^{-3}. While this appears to be a modest improvement compared to the differential evolution estimate, when the full model output is used to estimate the data, the R^2 value increases to a relative maximum of 9.9833×10^{-1} and the MSE reduces to a relative minimum of 6.4848×10^{-5}. The mean trajectory $\mu(t)$ had an average normalized band width of 9.4% (max 23.5%), and the empirical coverage of observed test points within these intervals was 100.0%, indicating that the model provides well-calibrated local uncertainty estimates.

The chemotherapy-treated cohort that experienced partial remission can be seen in Fig. 4b, grouped together with the differential evolution and PINN models that fit the data. The differential evolution results in an R^2 value of 9.5259×10^{-1} with an MSE of 1.8301×10^{-3}. Using the output of the PINN estimation of the parameter values with the SciPy ODE numerical solver results in a slightly diminished R^2 value of 9.4968×10^{-1} and a higher MSE of 1.9426×10^{-3}. The PINN model has the greatest fit of the data, with the highest R^2 value of 9.9898×10^{-1} and the lowest MSE of

Parameter	Immunotherapy		Chemotherapy	
	PINN	**DE**	**PINN**	**DE**
r_t	0.55968	0.57247	0.56375	0.54967
b_t	8.70×10^{-7}	8.63×10^{-7}	9.29×10^{-7}	9.33×10^{-7}
α_{ti}	1.61×10^{-7}	1.61×10^{-7}	1.47×10^{-7}	1.49×10^{-7}
ϵ_t	—	—	0.04204	0.04350
τ_ϵ	—	—	0.02926	0.02918
ω_ϵ	—	—	8.71435	8.71435
η_i	0.01229	0.0129	—	—
τ_η	0.02976	0.02974	—	—
ω_η	3.51577	3.51577	—	—
σ_i	1995.7	2050	3631.0	3610
α_{it}	2.11×10^{-8}	2.46×10^{-8}	8.05×10^{-12}	8.05×10^{-12}
β_{it}	1.32×10^{-8}	1.12×10^{-8}	1.49×10^{-9}	1.49×10^{-9}
δ_i	0.00093	0.00090	0.00159	0.00160
C_ϵ	—	—	0.10320	0.17512
C_η	0.04381	0.10428	—	—

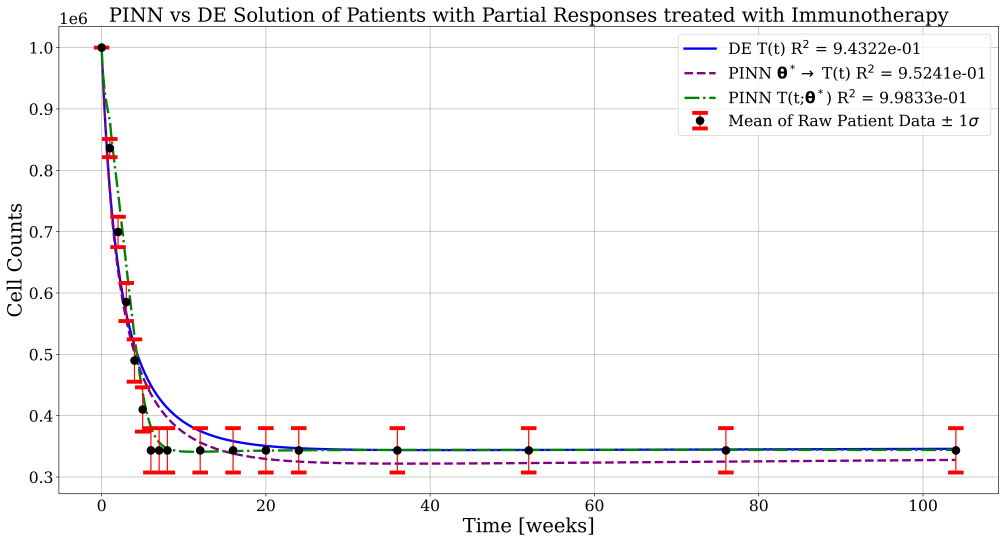
Table 6. Comparison of estimated parameters used in Eqs. (3) – (8) from PINN and Differential Evolution for patients who had a partial response undergoing treatment with immunotherapy or chemotherapy.

Metric	Immunotherapy			Chemotherapy		
	PINN $\widehat{T}(t;\boldsymbol{\theta}^*)$	**PINN** $\boldsymbol{\theta}^* \to T(t)$	**DE**	**PINN** $\widehat{T}(t;\boldsymbol{\theta}^*)$	**PINN** $\boldsymbol{\theta}^* \to T(t)$	**DE**
R^2	9.9833×10^{-1}	9.5241×10^{-1}	9.4322×10^{-1}	9.9898×10^{-1}	9.0697×10^{-1}	9.4488×10^{-1}
MSE	6.4848×10^{-5}	1.8444×10^{-3}	2.2005×10^{-3}	3.9478×10^{-5}	3.5911×10^{-3}	2.1277×10^{-3}

Table 7. Comparison of R^2 and MSE values for PINN and DE methods on immunotherapy and chemotherapy datasets for patients that experienced a partial response with therapy.

3.9478×10^{-5}. The mean trajectory $\mu(t)$ had an average normalized band width of 7.1% (max 17.9%), and the empirical coverage of observed test points within these intervals was 100.0%, indicating that the model provides well-calibrated local uncertainty estimates.

A consistent pattern across cohorts is that parameter estimates produced by the PINN do not always translate to improved performance when evaluated solely with a conventional numerical integrator. This discrepancy arises from the differing objectives: differential evolution directly minimizes a data-only loss using the numerical integrator, whereas the PINN jointly optimizes network weights and parameters under a composite loss that balances data fidelity, ODE residuals, and initialization constraints. The PINN's flexible function approximator can partially absorb model–data mismatch, thereby allowing parameter values that reduce the combined PINN loss but are not the minimizers of the integrator-only MSE. Consequently, in some cohorts the PINN-derived parameters produce superior integrator-based fits, while in others they do not — a result that reflects the multi-objective nature of PINN



(a) Comparison of the differential evolution results using SciPy, PINN parameter estimations with SciPy's ODE solver, and PINN parameter estimations with PINN's ODE solution plotted alongside a test dataset of patient tumor cell counts treated with immunotherapy in the partial response category.

training rather than a failure of parameter estimation per se. We therefore caution against treating a single-objective numerical solver (e.g., SciPy's **solve_ivp**) as a full cross-validation of PINN parameters unless the PINN training explicitly enforces equivalence between the network solution and the numerical integrator (for example, by embedding the integrator within the training loop or by placing stronger weight on the physics term), which was not the case for our training loop.

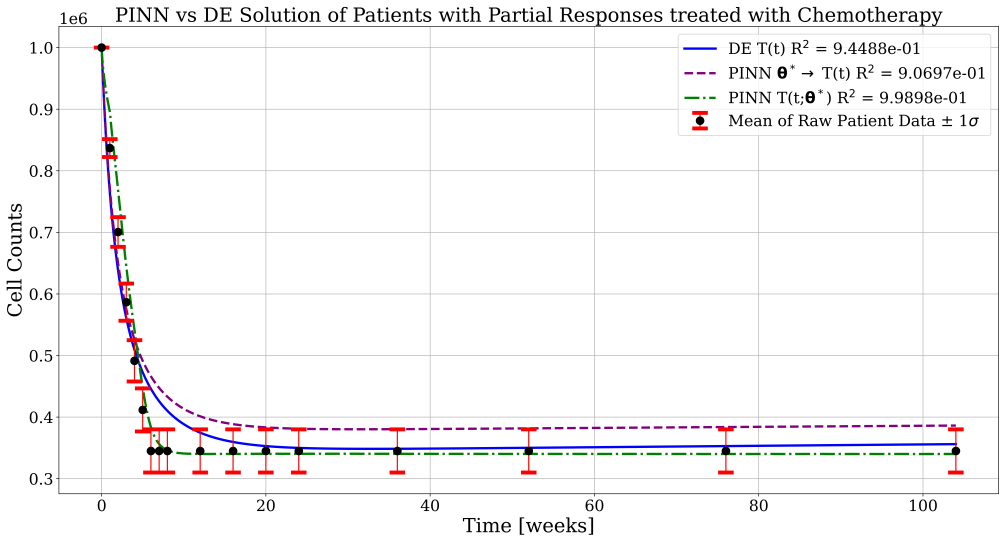
Conclusion

In this work, we proposed a dynamical model of tumor–immune interactions with therapeutic input terms that capture the build-up and decay of drug action through a convolutional dosing scheme, closely mimicking clinical treatment schedules.

To address the scarcity of cellular-resolution clinical data, we introduced a Monte Carlo-based algorithm to synthesize realistic longitudinal datasets conditioned on patient outcomes. This approach opens new possibilities for generating data-driven models in oncology, where direct cellular measurements are limited.

We then applied differential evolution (DE) for parameter estimation, providing a strong baseline for data-driven model fitting. These estimates were used as initialization for a physics-informed neural network (PINN), which jointly optimized parameters and trajectories under a composite loss function balancing data fidelity, ODE residuals, and initialization constraints. To improve training stability, we adapted a decorrelation metric from maximum-likelihood estimation and proposed it as a novel stopping criterion, showing that training could be halted once residual correlations diminished across epochs.

Our results demonstrate that PINNs can serve as a powerful alternative to traditional optimization-based parameter estimation. In low-variance datasets, PINN-derived parameters produced lower MSE and higher R^2 than DE when evaluated with a single-objective solver, though the neural solution itself exhibited



(b) Comparison of the differential evolution results using SciPy, PINN parameter estimations with SciPy's ODE solver, and PINN parameter estimations with PINN's ODE solution plotted alongside a test dataset of patient tumor cell counts treated with chemotherapy in the partial response category.

Fig 4. Comparison of differential evolution and PINN parameter and model estimations for tumor cell counts in partial response categories for immunotherapy and chemotherapy treatments.

modest degradation. In higher-variance datasets, PINN parameters had mixed performance under solver evaluation, but the full PINN model consistently outperformed both baselines, achieving substantial reductions in MSE and increases in R^2. These findings underscore a key distinction: while DE identifies parameters that minimize solver-based error, the PINN optimizes a multi-objective loss in which the network itself can absorb part of the model-data mismatch.

When analyzing the confidence intervals of the Monte Carlo-generated trajectories, we observed that low-variance trajectories often yielded low empirical coverage, whereas high-variance trajectories consistently achieved perfect empirical coverage. This discrepancy is likely due to the use of a single, fixed dropout rate across datasets. Achieving both well-calibrated empirical coverage for the Monte Carlo trajectories and accurate coverage of the mean data points may require tuning the dropout rate on a per-dataset basis during training and evaluation.

In summary, this work bridges mathematical modeling, machine learning, and cancer research by (i) proposing a realistic therapeutic model, (ii) introducing a novel data synthesis pipeline, (iii) adapting a decorrelation-based stopping criterion for PINN training, and (iv) benchmarking DE and PINN methods across treatment cohorts. The approach highlights both the promise and limitations of PINNs for biomedical parameter estimation: they offer strong data fits and flexibility, but their parameter values are not always directly transferable to single-objective numerical solvers. Future research should focus on integrating single-objective solver-based objectives into PINN training and applying the method to other clinical datasets to evaluate its translational potential.

Specifically, to address the computational demands and "parameter transferability" issues noted in this study, ongoing work extends the methodologies developed here into two distinct optimization frameworks. First, we are incorporating the

decorrelation-based early stopping criterion as a trigger for gradient-informed polishing within a novel GPU-accelerated Differential Evolution algorithm. This hybrid approach aims to bridge the gap between global search and local refinement, using the PINN's stability metrics to decide when to transition from exploration to gradient-based exploitation.

Furthermore, the structural complexity of these tumor–immune models suggests a need for optimization strategies that can handle both nonconvexity and specific structural constraints. To this end, we are currently expanding our optimization suite through the development of a GPU-accelerated Population-Based Great Deluge (PBGD) algorithm. By applying PBGD to both black-box and structure-aware optimization problems, we aim to further enhance the scalability and robustness of parameter estimation in complex biological systems, leveraging high-performance computing to achieve high-fidelity fits across even larger, more heterogeneous patient cohorts.

Acknowledgments

The views and opinions expressed in this article are those of the authors and do not necessarily reflect the official views or positions of PeopleTec. The authors thank the PeopleTec Technical Fellows for providing a review of our paper prior to submission.

References

1. Bray F, Laversanne M, Sung H, Ferlay J, Siegel RL, Soerjomataram I, et al. Global cancer statistics 2022: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA: a cancer journal for clinicians. 2024;74(3):229-63.

2. Kamat AM, Hahn NM, Efstathiou JA, Lerner SP, Malmström PU, Choi W, et al. Bladder cancer. The Lancet. 2016;388(10061):2796-810.

3. Sanli O, Dobruch J, Knowles MA, Burger M, Alemozaffar M, Nielsen ME, et al. Bladder cancer. Nature reviews Disease primers. 2017;3(1):1-19.

4. Shen Z, Shen T, Wientjes MG, O'Donnell MA, Au JLS. Intravesical treatments of bladder cancer. Pharmaceutical research. 2008;25(7):1500-10.

5. Shelley MD, Jones G, Cleves A, Wilt TJ, Mason MD, Kynaston HG. Intravesical gemcitabine therapy for non-muscle invasive bladder cancer (NMIBC): a systematic review. BJU international. 2012;109(4):496-505.

6. Chou R, Selph S, Buckley DI, Fu R, Griffin JC, Grusing S, et al. Intravesical therapy for the treatment of nonmuscle invasive bladder cancer: a systematic review and meta-analysis. The Journal of urology. 2017;197(5):1189-99.

7. Messing EM, Tangen CM, Lerner SP, Sahasrabudhe DM, Koppie TM, Wood DP, et al. Effect of intravesical instillation of gemcitabine vs saline immediately following resection of suspected low-grade non–muscle-invasive bladder cancer on tumor recurrence: SWOG S0337 randomized clinical trial. Jama. 2018;319(18):1880-8.

8. Holzbeierlein JM, Bixler BR, Buckley DI, Chang SS, Holmes R, James AC, et al. Diagnosis and treatment of non-muscle invasive bladder cancer: AUA/SUO guideline: 2024 amendment. The Journal of urology. 2024;211(4):533-8.

9. Griffiths TL, on Bladder Cancer A. Current perspectives in bladder cancer management. International journal of clinical practice. 2013;67(5):435-48.

10. Babjuk M, Burger M, Compérat EM, Gontero P, Mostafid AH, Palou J, et al. European association of urology guidelines on non-muscle-invasive bladder cancer (TaT1 and carcinoma in situ)-2019 update. European urology. 2019;76(5):639-57.

11. Gontero P, Birtle A, Capoun O, Compérat E, Dominguez-Escrig JL, Liedberg F, et al. European Association of Urology guidelines on non–muscle-invasive bladder cancer (TaT1 and carcinoma in situ)—a summary of the 2024 guidelines update. European Urology. 2024;86(6):531-49.

12. Chang SS, Bochner BH, Chou R, Dreicer R, Kamat AM, Lerner SP, et al. Treatment of non-metastatic muscle-invasive bladder cancer: AUA/ASCO/ASTRO/SUO guideline. The Journal of urology. 2017;198(3):552-9.

13. Witjes JA, Bruins HM, Cathomas R, Compérat EM, Cowan NC, Gakis G, et al. European Association of Urology guidelines on muscle-invasive and metastatic bladder cancer: summary of the 2020 guidelines. European urology. 2021;79(1):82-104.

14. von der Maase H, Hansen S, Roberts J, Dogliotti L, Oliver T, Moore M, et al. Gemcitabine and cisplatin versus methotrexate, vinblastine, doxorubicin, and cisplatin in advanced or metastatic bladder cancer: results of a large, randomized, multinational, multicenter, phase III study. Journal of clinical oncology. 2000;18(17):3068-77.

15. Sternberg CN, De Mulder P, Schornagel J, Théodore C, Fossa S, Van Oosterom A, et al. Randomized phase III trial of high–dose-intensity methotrexate, vinblastine, doxorubicin, and cisplatin (MVAC) chemotherapy and recombinant human granulocyte colony-stimulating factor versus classic MVAC in advanced urothelial tract tumors: European Organization for Research and Treatment of Cancer Protocol No. 30924. Journal of Clinical Oncology. 2001;19(10):2638-46.

16. Von Der Maase H, Sengelov L, Roberts JT, Ricci S, Dogliotti L, Oliver T, et al. Long-term survival results of a randomized trial comparing gemcitabine plus cisplatin, with methotrexate, vinblastine, doxorubicin, plus cisplatin in patients with bladder cancer. Journal of clinical oncology. 2005;23(21):4602-8.

17. Bellmunt J, De Wit R, Vaughn DJ, Fradet Y, Lee JL, Fong L, et al. Pembrolizumab as second-line therapy for advanced urothelial carcinoma. New England Journal of Medicine. 2017;376(11):1015-26.

18. Wołącewicz M, Hrynkiewicz R, Grywalska E, Suchojad T, Leksowski T, Roliński J, et al. Immunotherapy in bladder cancer: current methods and future perspectives. Cancers. 2020;12(5):1181.

19. Balar AV, Kamat AM, Kulkarni GS, Uchio EM, Boormans JL, Roumiguié M, et al. Pembrolizumab monotherapy for the treatment of high-risk non-muscle-invasive bladder cancer unresponsive to BCG (KEYNOTE-057): an open-label, single-arm, multicentre, phase 2 study. The Lancet Oncology. 2021;22(7):919-30.

20. Dobruch J, Oszczudłowski M. Bladder cancer: current challenges and future directions. Medicina. 2021;57(8):749.

21. Shiravand Y, Khodadadi F, Kashani SMA, Hosseini-Fard SR, Hosseini S, Sadeghirad H, et al. Immune checkpoint inhibitors in cancer therapy. Current Oncology. 2022;29(5):3044-60.

22. Dyrskjøt L, Hansel DE, Efstathiou JA, Knowles MA, Galsky MD, Teoh J, et al. Bladder cancer. Nature reviews Disease primers. 2023;9(1):58.

23. Necchi A, Roumiguié M, Kamat AM, Shore ND, Boormans JL, Esen AA, et al. Pembrolizumab monotherapy for high-risk non-muscle-invasive bladder cancer without carcinoma in situ and unresponsive to BCG (KEYNOTE-057): a single-arm, multicentre, phase 2 trial. The Lancet Oncology. 2024;25(6):720-30.

24. Apolo AB, Ballman KV, Sonpavde G, Berg S, Kim WY, Parikh R, et al. Adjuvant pembrolizumab versus observation in muscle-invasive urothelial carcinoma. New England Journal of Medicine. 2025;392(1):45-55.

25. Powles T, Rosenberg JE, Sonpavde GP, Loriot Y, Durán I, Lee JL, et al. Enfortumab vedotin in previously treated advanced urothelial carcinoma. New England Journal of Medicine. 2021;384(12):1125-35.

26. Powles T, Valderrama BP, Gupta S, Bedke J, Kikuchi E, Hoffman-Censits J, et al. Enfortumab vedotin and pembrolizumab in untreated advanced urothelial cancer. New England Journal of Medicine. 2024;390(10):875-88.

27. Van Der Heijden MS, Sonpavde G, Powles T, Necchi A, Burotto M, Schenker M, et al. Nivolumab plus gemcitabine–cisplatin in advanced urothelial carcinoma. New England Journal of Medicine. 2023;389(19):1778-89.

28. Bunimovich-Mendrazitsky S, Shochat E, Stone L. Mathematical model of BCG immunotherapy in superficial bladder cancer. Bulletin of Mathematical Biology. 2007;69(6):1847-70.

29. Bunimovich-Mendrazitsky S, Byrne H, Stone L. Mathematical model of pulsed immunotherapy for superficial bladder cancer. Bulletin of mathematical biology. 2008;70(7):2055-76.

30. Bunimovich-Mendrazitsky S, Gluckman JC, Chaskalovic J. A mathematical model of combined bacillus Calmette-Guerin (BCG) and interleukin (IL)-2 immunotherapy of superficial bladder cancer. Journal of theoretical biology. 2011;277(1):27-40.

31. Bunimovich-Mendrazitsky S, Goltser Y. Use of quasi-normal form to examine stability of tumor-free equilibrium in a mathematical model of BCG treatment of bladder cancer. Mathematical biosciences and engineering: MBE. 2011;8(2):529-47.

32. Bunimovich-Mendrazitsky S, Halachmi S, Kronik N. Improving Bacillus Calmette-Guérin (BCG) immunotherapy for bladder cancer by adding interleukin 2 (IL-2): a mathematical model. Mathematical medicine and biology: a journal of the IMA. 2016;33(2):159-88.

33. Yosef M, Bunimovich-Mendrazitsky S. Mathematical model of MMC chemotherapy for non-invasive bladder cancer treatment. Frontiers in Oncology. 2024;14:1352065.

34. Storn R, Price K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization. 1997;11(4):341-59.

35. Storn R. Differrential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical report, International Computer Science Institute. 1995;11.

36. Price KV. Differential evolution: a fast and simple numerical optimizer. In: Proceedings of North American fuzzy information processing. IEEE; 1996. p. 524-7.

37. Price KV, Storn RM, Lampinen JA. Differential evolution: a practical approach to global optimization. Springer; 2005.

38. Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics. 2019;378:686-707. Available from: `https://www.sciencedirect.com/science/article/pii/S0021999118307125`. doi:https://doi.org/10.1016/j.jcp.2018.10.045.

39. Yazdani A, Lu L, Raissi M, Karniadakis GE. Systems biology informed deep learning for inferring parameters and hidden dynamics. PLoS computational biology. 2020;16(11):e1007575.

40. Cuomo S, Di Cola VS, Giampaolo F, Rozza G, Raissi M, Piccialli F. Scientific machine learning through physics–informed neural networks: Where we are and what's next. Journal of Scientific Computing. 2022;92(3):88.

41. Raissi M, Perdikaris P, Ahmadi N, Karniadakis GE. Physics-informed neural networks and extensions. arXiv preprint arXiv:240816806. 2024.

42. Sakaguchi S, Yamaguchi T, Nomura T, Ono M. Regulatory T cells and immune tolerance. cell. 2008;133(5):775-87.

43. Nagaraj S, Gupta K, Pisarev V, Kinarsky L, Sherman S, Kang L, et al. Altered recognition of antigen is a mechanism of CD8+ T cell tolerance in cancer. Nature medicine. 2007;13(7):828-35.

44. Duraiswamy J, Kaluza KM, Freeman GJ, Coukos G. Dual blockade of PD-1 and CTLA-4 combined with tumor vaccine effectively restores T-cell rejection function in tumors. Cancer research. 2013;73(12):3591-603.

45. Nurieva R, Wang J, Sahoo A. T-cell tolerance in cancer. Immunotherapy. 2013;5(5):513-31.

46. Gabrilovich DI, Nagaraj S. Myeloid-derived suppressor cells as regulators of the immune system. Nature reviews immunology. 2009;9(3):162-74.

47. Wherry EJ. T cell exhaustion. Nature immunology. 2011;12(6):492-9.

48. Glick A, Mastroberardino A. An optimization method for treating solid tumors with combined therapy using the Great Deluge algorithm. Discrete and Continuous Dynamical Systems - B. 2024;29(3):1525-48. doi:10.3934/dcdsb.2023143.

49. Eisenhauer EA, Therasse P, Bogaerts J, Schwartz LH, Sargent D, Ford R, et al. New response evaluation criteria in solid tumours: Revised RECIST guideline (version 1.1). European Journal of Cancer. 2009;45(2):228-47. doi:10.1016/j.ejca.2008.10.026.

50. Glick A. Bladder Cancer Physics-Informed Neural Network (PINN) Simulation Code; 2026. Source code used to generate cell data and simulate tumor–immune dynamics and train physics-informed neural networks. Available from: `https://gitlab.com/glicka/bladder-cancer-pinn`.

51. Song D, Powles T, Shi L, Zhang L, Ingersoll MA, Lu YJ. Bladder cancer, a unique model to understand cancer immunity and develop immunotherapy approaches. The Journal of pathology. 2019;249(2):151-65.

52. Jadon A, Patil A, Jadon S. A Comprehensive Survey of Regression Based Loss Functions for Time Series Forecasting. arXiv preprint arXiv:221102989. 2022.

53. Park Y. Logarithmic Loss Transformation to Improve Anomaly Detection Model Training Stability. arXiv preprint arXiv:220105748. 2022.

54. Ren J, Zhang M, Yu C, Liu Z. Balanced MSE for Imbalanced Visual Regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022. p. 12582-91.

55. Glick A, Brubaker E, Cabrera-Palmer B, Gerling M, Boado ES, Vetter K. Characterization of a collimated neutron imager for low-rate fast neutron imaging. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. 2022;1026:166230. doi:https://doi.org/10.1016/j.nima.2021.166230.