

Do Biomedical Tasks Require Biomedical Foundation Models?

This manuscript ([permalink](#)) was automatically generated from [AdaptInfer/fm-survey@5b5baaf](#) on March 1, 2026.

Authors

- **Jingyun Jia**

 [0009-0006-3241-3485](#) ·  [Clouddelta](#)

Department of Statistics, University of Wisconsin-Madison

- **Zhiyuan Li**

 [0009-0006-6016-7381](#) ·  [LZYEIL](#)

Department of Computer Sciences, University of Wisconsin-Madison

- **Ben Lengerich** 

 [0000-0001-8690-9554](#) ·  [blengerich](#) ·  [ben_lengerich](#)

Department of Statistics, University of Wisconsin-Madison

✉ — Correspondence possible via [GitHub Issues](#) or email to Ben Lengerich <lengerich@wisc.edu>.

Abstract

Foundation models are increasingly used across biomedical domains, including clinical text, medical imaging, genomics, and protein modeling. At the same time, rapidly advancing general-purpose foundation models can often be adapted to biomedical tasks through prompting, fine-tuning, and tool use. This raises a central question: **do biomedical applications require domain-specific foundation models, or can general models be adapted effectively?**

In this review, we examine the landscape of biomedical foundation models and compare domain-specific pretraining with adaptation of general models. We discuss how these approaches interact with biomedical data systems and the challenges of evaluation, reliability, and deployment. We conclude by outlining key open problems that will shape the future of foundation models in biomedical research and healthcare.

Introduction

The Current State of Domain-Specific Biomedical Foundation Models

Overview of foundation models trained directly on biomedical data across major modalities such as clinical text, imaging, genomics, proteins, and EHRs.

Adapting General Foundation Models to Biomedical Tasks

How general-purpose foundation models (e.g., large language and vision models) are adapted to biomedical applications through prompting, fine-tuning, and tool use.

Integrating Foundation Models with Biomedical Data Systems

How foundation models interact with structured biomedical data and knowledge sources, including electronic health records, ontologies, and databases.

Evaluation, Reliability, and Deployment of Biomedical Foundation Models

Challenges in deploying and evaluating biomedical foundation models and ensuring reliability, including benchmarking, dataset bias, interpretability, and clinical validation.

Open Problems and Future Directions

Key unresolved questions and research opportunities that will shape the development and deployment of foundation models in biomedicine.

Conclusions

References

Appendix A

This appendix gives full proofs for Proposition 1 and Corollary 1. We keep the weighted support-set notation from the Introduction and make all linear-algebra steps explicit.

A.0 Preliminaries and identities

- **Joint features.** For any pair (x, c) , define

$$\psi(x, c) := x \otimes \phi(c) \in \mathbb{R}^{d_x d_c}.$$

For each indexed training example a (standing in for (i, j)), write $\psi_a := \psi(x_a, c_a)$.

- **Design/labels/weights.** Stack $N = \sum_i m_i$ training rows:

$$Z \in \mathbb{R}^{N \times d_x d_c} \text{ with rows } Z_a = \psi_a^T, \quad y \in \mathbb{R}^N, \quad W = \text{diag}(w_a) \in \mathbb{R}^{N \times N}, \quad w_a \geq 0.$$

Define the (unweighted) Gram matrix $K := ZZ^\top$ and the weighted Gram

$$K_W := W^{1/2} K W^{1/2} = W^{1/2} Z Z^\top W^{1/2}.$$

For a query (x, c) , let $k(\cdot, (x, c)) := Z \psi(x, c) \in \mathbb{R}^N$ and $k_{(x, c)} := W^{1/2} k(\cdot, (x, c))$.

- **Vectorization identity.** For conformable matrices A, B, C ,

$$\text{vec}(ABC) = (C^\top \otimes A) \text{vec}(B), \quad \langle \text{vec}(B), x \otimes z \rangle = x^\top B z.$$

- **Weighted ridge solution.** For any $X \in \mathbb{R}^{N \times p}$, ridge objective

$$\min_{\beta} \|W^{1/2}(y - X\beta)\|_2^2 + \lambda \|\beta\|_2^2$$

has unique minimizer $\hat{\beta} = (X^\top W X + \lambda I)^{-1} X^\top W y$ and equivalent dual form

$$\hat{\beta} = X^\top W^{1/2} (W^{1/2} X X^\top W^{1/2} + \lambda I)^{-1} W^{1/2} y.$$

Predictions for a new feature vector x_\star equal

$$\hat{f}(x_\star) = x_\star^\top \hat{\beta} = \underbrace{(W^{1/2} X x_\star)^\top}_{k_\star^\top} (W^{1/2} X X^\top W^{1/2} + \lambda I)^{-1} W^{1/2} y.$$

This is **kernel ridge regression** (KRR) with kernel $K_W = W^{1/2} X X^\top W^{1/2}$ and query vector $k_\star = W^{1/2} X x_\star$.

A.1 Proof of Proposition 1(A): explicit varying-coefficients \Leftrightarrow weighted KRR on joint features

Assume the linear, squared-loss setting with $y = \langle \theta(c), x \rangle + \varepsilon$ and $\mathbb{E}[\varepsilon] = 0$.

Let the varying-coefficients model be $\theta(c) = B \phi(c)$ with $B \in \mathbb{R}^{d_x \times d_c}$ and ridge penalty $\lambda \|B\|_F^2$.

Step 1 (reduce to ridge in joint-feature space).

Vectorize B as $\beta = \text{vec}(B) \in \mathbb{R}^{d_x d_c}$.

By the identity above,

$$x_a^\top B \phi(c_a) = \langle \beta, x_a \otimes \phi(c_a) \rangle = \langle \beta, \psi_a \rangle.$$

Thus the weighted objective specialized from (\star) is

$$\min_{\beta \in \mathbb{R}^{d_x d_c}} \|W^{1/2}(y - Z\beta)\|_2^2 + \lambda \|\beta\|_2^2,$$

which is exactly weighted ridge with design $X \equiv Z$.

Step 2 (closed form and prediction).

By the ridge solution,

$$\hat{\beta} = (Z^T W Z + \lambda I)^{-1} Z^T W y,$$

and the prediction at a query (x, c) with joint feature $\psi(x, c)$ is

$$\hat{y}(x, c) = \psi(x, c)^T \hat{\beta} = \underbrace{(W^{1/2} Z \psi(x, c))^T}_{k_{(x,c)}} (W^{1/2} Z Z^T W^{1/2} + \lambda I)^{-1} W^{1/2} y.$$

Step 3 (kernel form).

Since $K := Z Z^T$ and $K_W := W^{1/2} K W^{1/2}$,

$$\hat{y}(x, c) = k_{(x,c)}^T (K_W + \lambda I)^{-1} W^{1/2} y.$$

Moreover, the (a, b) -th entry of the kernel matrix K is

$$K_{ab} = \langle \psi_a, \psi_b \rangle = \langle x_a \otimes \phi(c_a), x_b \otimes \phi(c_b) \rangle = \langle x_a, x_b \rangle \cdot \langle \phi(c_a), \phi(c_b) \rangle,$$

so (A) is precisely **KRR on joint features** with sample weights W .

This proves part (A). ■

A.2 Proof of Proposition 1(B): linear ICL \Rightarrow kernel regression

We analyze a single attention layer operating on the weighted support set $\mathcal{S}(c)$, using **linear** maps for queries, keys, and values:

$$q(x, c) = Q \psi(x, c), \quad k_a = K \psi_a, \quad v_a = V \psi_a,$$

with $Q \in \mathbb{R}^{d_q \times d_\psi}$, $K \in \mathbb{R}^{d_k \times d_\psi}$, $V \in \mathbb{R}^{d_v \times d_\psi}$, $d_\psi = d_x d_c$. Let the **unnormalized** attention score for index a be

$$s_a(x, c) := w_a \langle q(x, c), k_a \rangle = w_a \psi(x, c)^T Q^T K \psi_a.$$

Define normalized weights $\alpha_a(x, c) := s_a(x, c) / \sum_b s_b(x, c)$ (or any fixed positive normalization; the form below is pointwise in $\{\alpha_a\}$). The context representation and scalar prediction are

$$z(x, c) = \sum_a \alpha_a(x, c) v_a, \quad \hat{y}(x, c) = u^T z(x, c).$$

We prove two statements: **(B1)** exact KRR if the attention maps are fixed and only the readout is trained, and **(B2)** kernel regression with the NTK if the attention parameters are trained in the linearized regime.

A.2.1 (B1) Fixed attention, trained linear head \Rightarrow exact KRR

Assume Q, K, V are fixed functions (pretrained or chosen a priori), hence $\alpha_a(x, c)$ are **deterministic** functions of (x, c) and the support set. Define the induced **feature map**

$$\varphi(x, c) := \sum_a \alpha_a(x, c) v_a \in \mathbb{R}^{d_v}.$$

Stack $\varphi_a := \varphi(x_a, c_a)$ row-wise into $\Phi \in \mathbb{R}^{N \times d_v}$. Training only the readout u with weighted ridge,

$$\hat{u} \in \arg \min_u \|W^{1/2}(y - \Phi u)\|_2^2 + \lambda \|u\|_2^2$$

yields $\hat{u} = (\Phi^T W \Phi + \lambda I)^{-1} \Phi^T W y$ and predictions

$$\hat{y}(x, c) = \varphi(x, c)^T \hat{u} = \underbrace{(W^{1/2} \Phi \varphi(x, c))}_{k_{(x, c)}} (W^{1/2} \Phi \Phi^T W^{1/2} + \lambda I)^{-1} W^{1/2} y.$$

Therefore,

$$\boxed{\hat{y}(x, c) = k_{(x, c)}^T (K_W + \lambda I)^{-1} W^{1/2} y}, \quad K_W := W^{1/2} \underbrace{(\Phi \Phi^T)}_{=: K} W^{1/2},$$

which is exactly **kernel ridge regression** with kernel

$$k((x, c), (x', c')) = \langle \varphi(x, c), \varphi(x', c') \rangle.$$

Because $v_a = V \psi_a$ and $\alpha_a(x, c) \propto w_a \psi(x, c)^T Q^T K \psi_a$, φ is a linear transform of a **weighted average of joint features**; hence the kernel is a dot-product on linear transforms of $\{\psi_a\}$. This proves (B1). ■

A.2.2 (B2) Training attention in the linearized/NTK regime \Rightarrow kernel regression with NTK

Now let $\theta = (Q, K, V, u)$ be trainable, and suppose training uses squared loss with gradient flow (or sufficiently small steps) starting from initialization θ_0 . The **linearized model** around θ_0 is the first-order Taylor expansion

$$\hat{y}_\theta(x, c) \approx \hat{y}_{\theta_0}(x, c) + \nabla_{\theta} \hat{y}_{\theta_0}(x, c)^T (\theta - \theta_0) =: \hat{y}_{\theta_0}(x, c) + \phi_{\text{NTK}}(x, c)^T (\theta - \theta_0),$$

where $\phi_{\text{NTK}}(x, c) := \nabla_{\theta} \hat{y}_{\theta_0}(x, c)$ are the **tangent features**. Standard NTK results (for squared loss, gradient flow, and linearization-validity conditions) imply that the learned function equals **kernel regression with the NTK**:

$$k_{\text{NTK}}((x, c), (x', c')) := \langle \phi_{\text{NTK}}(x, c), \phi_{\text{NTK}}(x', c') \rangle,$$

i.e., predictions have the KRR form with kernel K_{NTK} on the training set (and explicit ridge if used, or implicit regularization via early stopping).

It remains to identify the structure of ϕ_{NTK} for our **linear attention** block and show it lies in the span of **linear transforms of joint features**. Differentiating $\hat{y}(x, c) = u^T \sum_a \alpha_a(x, c) V \psi_a$ at θ_0 yields four groups of terms:

- **Readout path (u).** $\partial \hat{y} / \partial u = \sum_a \alpha_a(x, c) V \psi_a = \varphi_0(x, c)$. This is linear in $\{\psi_a\}$.
- **Value path (V).** $\partial \hat{y} / \partial V = \sum_a \alpha_a(x, c) u \psi_a^T$. This contributes terms of the form $(u \otimes I) \sum_a \alpha_a(x, c) \psi_a$, i.e., linear in $\{\psi_a\}$.
- **Query/key paths (Q, K).** For linear attention with scores $s_a = w_a \psi(x, c)^T Q^T K \psi_a$ and normalized $\alpha_a = s_a / \sum_b s_b$, derivatives of α_a w.r.t. Q and K are linear combinations of $\psi(x, c)$ and $\{\psi_a\}$:

$$\frac{\partial \alpha_a}{\partial Q} \propto \sum_b [\delta_{ab} - \alpha_b(x, c)] w_a w_b (K \psi_a \psi(x, c)^T),$$

$$\frac{\partial \alpha_a}{\partial K} \propto \sum_b [\delta_{ab} - \alpha_b(x, c)] w_a w_b (\psi(x, c) \psi_a^T Q^T),$$

and hence $\partial \hat{y} / \partial Q, \partial \hat{y} / \partial K$ are finite linear combinations of tensors each bilinear in $\psi(x, c)$ and some ψ_a . Contracting with u and V produces terms *linear* in $\psi(x, c)$ and linear in the set $\{\psi_a\}$.

Collecting all components, the tangent feature map can be written as

$$\phi_{\text{NTK}}(x, c) = \mathcal{L}(\psi(x, c), \{\psi_a\}),$$

where \mathcal{L} is a fixed linear operator determined by θ_0 , W , and the normalization rule for attention. Consequently, the NTK takes the **dot-product** form

$$k_{\text{NTK}}((x, c), (x', c')) = \Psi(x, c)^T \mathcal{M} \Psi(x', c'),$$

for some positive semidefinite matrix \mathcal{M} and a finite-dimensional feature stack Ψ that concatenates linear transforms of $\psi(x, c)$ and of the support-set $\{\psi_a\}$. In particular, k_{NTK} is a dot-product kernel on **linear transforms of the joint features** (possibly augmented by normalization-dependent combinations). Therefore, training the linear-attention ICL model in the linearized regime equals kernel regression with such a kernel—completing (B2). ■

Assumptions for A.2.2. Squared loss; gradient flow (or sufficiently small steps); initialization independent of the data; and a regime where the linearization error stays controlled over training (e.g., small learning rate, sufficient width/depth so that the NTK remains close to its initialization).

A.3 Proof of Corollary 1: retrieval/gating/weighting as kernel/measure choices

In both A.1 and A.2, predictions have the KRR form

$$\hat{y}(x, c) = k_{(x, c)}^T (K^\sharp + \lambda I)^{-1} \mu,$$

where K^\sharp is a positive semidefinite kernel matrix computed over the support set (e.g., $K_W = W^{1/2} Z Z^T W^{1/2}$ in A.1 or $W^{1/2} \Phi \Phi^T W^{1/2} / K_{\text{NTK}}$ in A.2), $k_{(x, c)}$ is the associated query vector, and $\mu = W^{1/2} y$ (or an equivalent reweighting).

- **Retrieval $R(c)$ / gating.** Changing the support set $S(c)$ (e.g., via a retriever or a gating policy) **removes or adds rows/columns** in K^\sharp and entries in $k_{(x, c)}$. This is equivalent to changing the **empirical measure** over which the kernel smoother is computed (i.e., which samples contribute and how).
- **Weights $w_{ij}(c)$.** Changing the weights modifies W and hence replaces K by $K_W = W^{1/2} K W^{1/2}$ and k by $k_{(x, c)} = W^{1/2} k$. This is standard **importance weighting** in kernel regression.
- **Induced kernels.** Attention, value projections, or learned encoders change the **feature map** (e.g., $\psi \mapsto V\psi$ or $\psi \mapsto Q\psi$), thereby changing the kernel $k((x, c), (x', c')) = \langle \Phi(x, c), \Phi(x', c') \rangle$.

Thus retrieval/gating instantiate **neighborhood selection** (measure choice), and value/query/key processing instantiate **kernel choice**. ■

A.4 Remarks

- **No Gaussianity is required.** Part (A) only uses squared loss and linear algebra; the noise model $y = f(x, c) + \varepsilon$ with $\mathbb{E}[\varepsilon] = 0$ suffices.
- **Early stopping vs. explicit ridge.** If training uses early stopping rather than explicit λ , the resulting predictor is still a kernel regressor with an *implicit* regularization parameter controlled by stopping time (for gradient flow on squared loss).
- **Multiple layers / nonlinear value stacks.** With deeper nonlinear stacks, the exact identities above become local/first-order (linearized) approximations; the NTK statement continues to apply under its usual conditions.

