Model details

Let $X \in \mathbb{R}_{\geq 0}^{p\times n}$ denote the nonnegative gene expression matrix with p genes and n subjects. DeSurv approximates $X \approx WH$ with $W \in \mathbb{R}_{\geq 0}^{p\times k}$ and $H \in \mathbb{R}_{\geq 0}^{k\times n}$, and links the shared program matrix W to survival via an elastic-net–penalized Cox model.

Let $y \in \mathbb{R}_{\geq 0}^{n}$ be the observed survival times, $\delta \in \{0,1\}^n$ the event indicators, and $Z = W^\top X \in \mathbb{R}^{k\times n}$ the sample-wise program loadings. We write Z_i for the ith column of Z, and $n_{\text{event}} = \sum_{i=1}^{n} \delta_i$. For convenience we recall the DeSurv loss from Equation~**??** in the main text and rewrite it as

$$\begin{aligned}\mathcal{L}(W,H,\beta) = \\ &(1-\alpha)\left(\frac{1}{2np}\|X - WH\|_F^2 + \frac{\lambda_H}{2nk}\|H\|_F^2\right) \\ &-\alpha\left(\frac{2}{n_{event}}\ell(W,\beta) - \lambda\{\xi\|\beta\|_1 + \tfrac{1-\xi}{2}\|\beta\|_2^2\}\right)\end{aligned}$$

where $\|\cdot\|_F$ is the Frobenius norm and $\beta \in \mathbb{R}^k$ are the Cox coefficients. The Cox log-partial likelihood $\ell(W,\beta)$ is

$$\ell(W,\beta) = \sum_{i=1}^{n} \delta_i \left[Z_i^\top \beta - \log\left(\sum_{j:y_j \geq y_i} \exp(Z_j^\top \beta)\right)\right]. \tag{1}$$

Hyperparameters satisfy $\alpha \in [0,1)$, $\lambda_H \geq 0$, $\lambda \geq 0$, and $\xi \in [0,1]$. The constants $1/(2np)$, $2/n_{\text{event}}$, and $1/(2nk)$ are chosen for numerical convenience and do not affect the minimizer.

Optimization Algorithm

Block coordinate descent scheme

Algorithm~1 summarizes the block coordinate descent (BCD) scheme used to minimize $\mathcal{L}(W,H,\beta)$. At each outer iteration, we update H, then W, then β while holding the other blocks fixed.

Algorithm 1 DeSurv block coordinate descent

```
Input: $X \in \mathbb{R}^{p \times n}_{\geq 0}$, survival times $y \in \mathbb{R}^{n}_{\geq 0}$, event indicators $\delta \in \{0,1\}^n$, tolerance $tol$, maximum iterations $maxit$, supervision parameter $\alpha$, rank $k$, penalties $\lambda_H$, $\lambda$, and $\xi$
Output: Fitted DeSurv parameters $(W, H, \beta)$
Initialize $W^{(0)}, H^{(0)}$ with positive entries (e.g., $W^{(0)}, H^{(0)} \sim \text{Uniform}(0, \max X)$)
Initialize $\beta^{(0)}$ (e.g., $\beta^{(0)} = \mathbf{1}$)
$loss = \mathcal{L}(W^{(0)}, H^{(0)}, \beta^{(0)})$
$eps = \infty$, $t = 0$
while $eps \geq tol$ and $t < maxit$ do
    (H-update)
        Update $H^{(t+1)}$ from $H^{(t)}$ using the multiplicative rule in Eq. 2,
        holding $W^{(t)}$ and $\beta^{(t)}$ fixed.
    (W-update)
        Update $W^{(t+1)}$ from $W^{(t)}$ using the hybrid multiplicative rule in Eq. 6,
        holding $H^{(t+1)}$ and $\beta^{(t)}$ fixed, and perform backtracking to ensure
        $\mathcal{L}$ does not increase.
    ($\beta$-update)
        Update $\beta^{(t+1)}$ from $\beta^{(t)}$ using a Newton-like step for the Cox loss
        (Eq. 7), holding $W^{(t+1)}$ and $H^{(t+1)}$ fixed.
    $lossNew = \mathcal{L}(W^{(t+1)}, H^{(t+1)}, \beta^{(t+1)})$
    $eps = |lossNew - loss|/|loss|$
    $loss = lossNew$
    $t = t + 1$
end while
return $\hat{W} = W^{(t+1)}, \hat{H} = H^{(t+1)}, \hat{\beta} = \beta^{(t+1)}$
```

Update for H

Conditional on W, the loss $\mathcal{L}(W, H, \beta)$ reduces to a strictly convex quadratic function of H. We adopt the standard NMF multiplicative update with ℓ_2 penalty:

$$H \leftarrow \max\left(H \odot \frac{W^\top X}{W^\top W H + \lambda_H H + \varepsilon_H}, \varepsilon_H\right), \tag{2}$$

where $\odot$ denotes elementwise multiplication, all divisions are elementwise, and $\varepsilon_H > 0$ is a small floor to prevent entries from becoming exactly zero. This update is equivalent to a majorization–minimization step and guarantees a nonincreasing reconstruction term conditional on W [@seung2001algorithms; @pascual-montano2006nonsmooth]. The ε_H-floor ensures that iterates remain in the interior of the nonnegative orthant, which simplifies the convergence analysis.

Update for W

For W, we construct a hybrid multiplicative update that balances the contributions of the NMF and Cox gradients. Let

$$\nabla_W \mathcal{L}_{\text{NMF}}(W, H) = \frac{1}{np}(WH - X)H^\top,$$

and let

$$\nabla_W \mathcal{L}_{\text{Cox}}(W, \beta) = \frac{2}{n_{event}} \nabla_W \ell(W, \beta).$$

where $\nabla_W \ell(W, \beta)$ denotes the Cox gradient with respect to W. A closed-form expression for $\nabla_W \ell$ is

$$\nabla_W \ell(W, \beta) = \sum_{i=1}^{n} \delta_i \left(x_i - \frac{\sum_{j:y_j \geq y_i} x_j \exp(x_j^\top W \beta)}{\sum_{j:y_j \geq y_i} \exp(x_j^\top W \beta)} \right) \beta^\top, \tag{3}$$

where x_i denotes the ith column of X.

We define the gradient-balancing factor

$$\delta^{(t)} = \frac{\left\|(1-\alpha)\nabla_W \mathcal{L}_{\mathrm{NMF}}(W^{(t)}, H^{(t+1)})\right\|_F^2}{\left\|\alpha\nabla_W \mathcal{L}_{Cox}(W^{(t)}, \beta^{(t)})\right\|_F^2}, \tag{4}$$

and clip $\delta^{(t)}$ to avoid extreme values: $\delta^{(t)} \leftarrow \min(\delta^{(t)}, \delta_{\max})$ with $\delta_{\max} = 10^6$ in all experiments.

The multiplicative factor for W is then

$$R^{(t)} = \frac{\frac{(1-\alpha)}{np} X H^{(t+1)\top} + \frac{2\alpha}{n_{\mathrm{event}}}\,\delta^{(t)}\,\nabla_W \ell(W^{(t)}, \beta^{(t)})}{\frac{(1-\alpha)}{np} W^{(t)} H^{(t+1)} H^{(t+1)\top} + \varepsilon_W}, \tag{5}$$

and the proposed update is

$$W^{(t+1)} = \max\Big(W^{(t)} \odot R^{(t)}, \varepsilon_W\Big)\,, \tag{6}$$

with a small floor $\varepsilon_W > 0$. When $\alpha = 0$, this reduces to the standard multiplicative update for NMF [@seung2001algorithms]; for $\alpha > 0$, the Cox gradient perturbs the update in a direction that decreases the supervised loss.

Because the combined multiplicative step does not, by itself, guarantee a nonincrease in the full loss $\mathcal{L}$, we embed it in a backtracking line search.

Algorithm 2 W update with backtracking

```
Input: Value of $W$ and the previous iteration $t$: $W^{(t)}$, the multiplicative update term at the current iteration
    $R^{(t+1)}$, the max number of backtracking updates $max\_bt$, the backtracking parameter $\rho \subset (0,1)$
Output: $W^{(t+1)}$
$\theta = 1$
b = 1
$flag\_accept = FALSE$
while $b \leq max\_bt$ do
    $W^{(t+1)}_{cand} = W^{(t)} \odot [(1-\theta) + \theta R^{(t+1)}]$
    (Column normalization)
        Compute $D = \mathrm{diag}\big(\|W^{(t+1)}_{(:,1)cand}\|_2, \ldots, \|W^{(t+1)}_{(:,k)cand}\|_2\big)$
        and set $W^{(t+1)}_{cand} \leftarrow W^{(t+1)}_{cand} D^{-1}$, $H^{(t+1)}_{cand} \leftarrow DH^{(t+1)}$, and $\beta^{(t)}_{cand} \leftarrow D\beta^{(t)}$
    if $\mathcal{L}(W^{(t+1)}_{cand}, H^{(t+1)}, \beta^{(t)}) \leq \mathcal{L}(W^{(t)}, H^{(t+1)}, \beta^{(t)})$ then
        $W^{(t+1)} = W^{(t+1)}_{cand}$
        $H^{(t+1)} = H^{(t+1)}_{cand}$
        $\beta^{(t)} = \beta^{(t)}_{cand}$
        $flag\_accept = TRUE$
        break
    end if
    $\theta = \theta * \rho$
    $b = b + 1$
end while
if $flag\_accept = FALSE$ then
    $W^{(t+1)} = W^{(t)}$
end if
```

The column normalization preserves both WH and $W\beta$, and therefore leaves the loss $\mathcal{L}$ invariant up to numerical error. Backtracking guarantees that the accepted W update does not increase $\mathcal{L}$.

Update for β

Conditional on (W, H), the loss in β reduces to a convex elastic-net–penalized Cox problem:

$$\min_{\beta \in \mathbb{R}^k} \left\{ -\frac{2\alpha}{n_{\text{event}}} \ell(W, \beta) + \lambda \left(\xi \|\beta\|_1 + \frac{1-\xi}{2} \|\beta\|_2^2 \right) \right\}.$$

We solve this subproblem by cyclic coordinate descent following [@simon2011regularization]. Writing $\ell(\beta) = \ell(W, \beta)$ and $\tilde{\eta}_i = Z_i^\top \beta$, the update for coordinate r has the closed form

$$\hat{\beta}_r = \frac{S\left(\frac{1}{n} \sum_{i=1}^n w(\tilde{\eta})_i z_{i,r} \left[v(\tilde{\eta})_i - \sum_{j \neq r} z_{i,j} \beta_j \right], \lambda \xi \right)}{\frac{1}{n} \sum_{i=1}^n w(\tilde{\eta})_i z_{i,r}^2 + \lambda(1-\xi)}, \tag{7}$$

where $S(a, \tau) = \text{sign}(a) \max(|a| - \tau, 0)$ is the soft-thresholding operator, and $w(\tilde{\eta})$, $v(\tilde{\eta})$ are standard local quadratic approximations of the Cox log-partial likelihood [@simon2011regularization]. We iterate coordinate updates until the subproblem converges to numerical tolerance, yielding a minimizer in β for the current W.

Normalization of W

After each accepted W update, we normalize the columns of W as $W \leftarrow WD^{-1}$, and adjust H and β as

$$H \leftarrow DH, \qquad \beta \leftarrow D\beta,$$

where D is the diagonal matrix of column ℓ_2-norms of W. This preserves the reconstruction WH and the linear predictor $W\beta$, leaving both the NMF and Cox terms in the loss unchanged. Normalization prevents degeneracy in the scale-nonidentifiable factorization and keeps columns of W comparable in magnitude and interpretable as gene programs.

Derivation of W update from projected coordinate descent

The W update can be derived directly from the projected coordinate descent update with a specific step size.

Recall that the overall loss function is

$$\mathcal{L}(W, H, \beta) = \frac{(1-\alpha)}{2np} ||X - WH||_F^2 - \frac{2\alpha}{n_{event}} \ell(W, \beta) + \lambda(\xi ||\beta||_1 + \frac{(1-\xi)}{2}, \tag{8}$$

where $\ell(W, \beta)$ is the log-partial likelihood for the Cox model. Let $\nabla_W \ell$ represent the derivative of $\ell(W, \beta)$ with respect to W. Then the derivative of the overall loss with respect to W is

$$\frac{\partial L}{\partial W} = \frac{(1-\alpha)}{np} \left(WHH^T - XH^T \right) - \frac{2\alpha}{n_{event}} \nabla_W \ell \tag{9}$$

Then the gradient descent update rule at iteration t is

$$\begin{aligned} W^{(t)} &= W^{(t-1)} - \gamma \left(\frac{\partial L}{\partial W} \right) \\ &= W^{(t-1)} - \gamma \left(\frac{(1-\alpha)}{np} \left(WHH^T - XH^T \right) - \frac{2\alpha}{n_{event}} \nabla_W \ell \right) \end{aligned} \tag{10}$$

$$\tag{11}$$

Let the step size γ be defined as in [@seung2001algorithms]:

$$\gamma = \frac{np}{(1-\alpha)} \frac{W}{WHH^T} \tag{12}$$

Then the update becomes

$$
\begin{aligned}
W^{(t)} &= W^{(t-1)} - \frac{np}{(1-\alpha)} \frac{W}{WHH^T} \left(\frac{(1-\alpha)}{np} \left(WHH^T - XH^T \right) - \frac{2\alpha}{n_{event}} \nabla_W \ell \right) \\
&= \frac{W}{WHH^T} XH^T + \frac{2\alpha np}{n_{event}(1-\alpha)} \nabla_W \ell \\
&= W \odot \frac{XH^T + \frac{2\alpha np}{n_{event}(1-\alpha)} \nabla_W \ell}{WHH^T} \\
&= W \odot \frac{\frac{(1-\alpha)}{np} XH^T + \frac{2\alpha}{n_{event}} \nabla_W \ell}{\frac{(1-\alpha)}{np} WHH^T}
\end{aligned} \tag{13}
$$

Finally, projected coordinate descent projects the W update in to the positive space

$$
W^{(t)} = \max \left(W \odot \frac{\frac{(1-\alpha)}{np} XH^T + \frac{2\alpha}{n_{event}} \nabla_W \ell}{\frac{(1-\alpha)}{np} WHH^T}, 0 \right) \tag{14}
$$

Since $W \geq 0$ this is equivalent to

$$
W^{(t)} = W \odot \max \left(\frac{\frac{(1-\alpha)}{np} XH^T + \frac{2\alpha}{n_{event}} \nabla_W \ell}{\frac{(1-\alpha)}{np} WHH^T}, 0 \right) \tag{15}
$$

which matches the multiplicative form used in the software implementation.

Convergence proof

We show that, under mild regularity conditions, the block coordinate descent (BCD) Algorithm~S1 converges to a stationary point of the DeSurv loss function

$$
\mathcal{L}(W, H, \beta).
$$

For clarity, we first analyze the algorithm *without* the column normalization of W inside the loop, and then argue in Section~ that the normalization step preserves stationarity of limit points.

Throughout, let $\theta = (W, H, \beta)$ and denote $\mathcal{L}(\theta) = \mathcal{L}(W, H, \beta)$. The feasible set is

$$
\Theta = \{(W, H, \beta) : W \in \mathbb{R}_{\geq 0}^{p \times k},\ H \in \mathbb{R}_{\geq 0}^{k \times n},\ \beta \in \mathbb{R}^k\}.
$$

Assumption 1 (Regularity and parameter space). We assume:

(i) The data $X \in \mathbb{R}_{\geq 0}^{p \times n}$, $y \in \mathbb{R}^n$, and $\delta \in \{0, 1\}^n$ are fixed and bounded.

(ii) The hyperparameters satisfy $\lambda_H > 0$, $\lambda > 0$, $\alpha \in [0, 1)$, and $\xi \in [0, 1]$.

(iii) The initial iterate $\theta^{(0)} = (W^{(0)}, H^{(0)}, \beta^{(0)})$ lies in Θ and satisfies $W^{(0)}, H^{(0)} > 0$ elementwise.

Assumption 2 (Block updates). At each outer iteration t, the three block updates in Algorithm~S1 satisfy:

(i) *H-update.* For fixed $(W^{(t)}, \beta^{(t)})$, the update $H^{(t)} \mapsto H^{(t+1)}$ is given by the multiplicative rule

$$
H \leftarrow H \odot \frac{W^\top X}{W^\top WH + \lambda_H H},
$$

applied at $(W^{(t)}, H^{(t)})$. This rule preserves nonnegativity and yields a nonincreasing value of the reconstruction term conditional on $W^{(t)}$ and $\beta^{(t)}$; see e.g. [@seung2001algorithms; @pascual-montano2006nonsmooth; @lin2007convergence].

(ii) *W-update.* For fixed $(H^{(t+1)}, \beta^{(t)})$, the update $W^{(t)} \mapsto W^{(t+1)}$ is the hybrid multiplicative step followed by backtracking as in Algorithm S2, producing $W^{(t+1)}$ such that

$$\mathcal{L}(W^{(t+1)}, H^{(t+1)}, \beta^{(t)}) \leq \mathcal{L}(W^{(t)}, H^{(t+1)}, \beta^{(t)}).$$

If no candidate satisfies the Armijo-type condition within the allowed number of backtracking steps, the algorithm sets $W^{(t+1)} := W^{(t)}$.

(iii) *β-update.* For fixed $(W^{(t+1)}, H^{(t+1)})$, the update $\beta^{(t)} \mapsto \beta^{(t+1)}$ is obtained by running the coordinate descent method of [@simon2011regularization] on the convex elastic-net Cox subproblem until convergence. Thus

$$\beta^{(t+1)} \in \arg\min_{\beta \in \mathbb{R}^k} \mathcal{L}(W^{(t+1)}, H^{(t+1)}, \beta),$$

and

$$\mathcal{L}(W^{(t+1)}, H^{(t+1)}, \beta^{(t+1)}) \leq \mathcal{L}(W^{(t+1)}, H^{(t+1)}, \beta^{(t)}).$$

Lemma 1 (Continuity and bounded level sets). Under Assumption 1, $\mathcal{L} : \Theta \to \mathbb{R}$ is continuous, and the initial sublevel set

$$\mathcal{S}_0 := \{\theta \in \Theta : \mathcal{L}(\theta) \leq \mathcal{L}(\theta^{(0)})\}$$

is nonempty, closed, and bounded.

Proof. The NMF term $\|X - WH\|_F^2$ is a polynomial in the entries of W and H, hence is continuously differentiable. The penalty $\|H\|_F^2$ is continuous as well. The Cox partial log-likelihood is a smooth function of the linear predictor $\eta_i = x_i^\top W\beta$, which is linear in (W, β), so this term is continuously differentiable in (W, β). The ℓ_2 penalty on β is smooth, and the ℓ_1 penalty is convex and lower semicontinuous. Therefore $\mathcal{L}$ is continuous on Θ.

The constraints $W, H \geq 0$ define closed convex cones, and $\beta \in \mathbb{R}^k$ is unconstrained. Because $\lambda_H > 0$ and $\lambda > 0$, the terms $\frac{\lambda_H}{nk}\|H\|_F^2$ and $\lambda\frac{1-\xi}{2}\|\beta\|_2^2$ dominate the objective as $\|H\|_F \to \infty$ or $\|\beta\|_2 \to \infty$, respectively. Thus the sublevel set $\mathcal{S}_0$ is bounded in (H, β). Since $W \geq 0$ and X is fixed, the reconstruction term and Cox term being bounded prevent W from diverging while $\mathcal{L}$ remains below $\mathcal{L}(\theta^{(0)})$, so W is bounded on $\mathcal{S}_0$. Because Θ is closed and $\mathcal{L}$ is continuous, $\mathcal{S}_0$ is closed. Nonemptiness follows from $\theta^{(0)} \in \mathcal{S}_0$. □

Lemma 2 (Monotone descent and existence of limit points). Under Assumptions 1 and 2, Algorithm~S1 (without W normalization) generates a sequence $\{\theta^{(t)}\}_{t \geq 0} \subset \mathcal{S}_0$ such that

$$\mathcal{L}(\theta^{(t+1)}) \leq \mathcal{L}(\theta^{(t)}) \quad \forall\, t,$$

and $\{\mathcal{L}(\theta^{(t)})\}$ converges to a finite limit $\mathcal{L}^*$. Moreover, $\{\theta^{(t)}\}$ is bounded and therefore admits at least one limit point.

Proof. By Assumption 2(i),

$$\mathcal{L}(W^{(t)}, H^{(t+1)}, \beta^{(t)}) \leq \mathcal{L}(W^{(t)}, H^{(t)}, \beta^{(t)}).$$

By Assumption 2(ii),

$$\mathcal{L}(W^{(t+1)}, H^{(t+1)}, \beta^{(t)}) \leq \mathcal{L}(W^{(t)}, H^{(t+1)}, \beta^{(t)}).$$

By Assumption 2(iii),

$$\mathcal{L}(W^{(t+1)}, H^{(t+1)}, \beta^{(t+1)}) \leq \mathcal{L}(W^{(t+1)}, H^{(t+1)}, \beta^{(t)}).$$

Combining these inequalities gives

$$\mathcal{L}(\theta^{(t+1)}) \leq \mathcal{L}(\theta^{(t)}) \quad \forall t,$$

so $\{\mathcal{L}(\theta^{(t)})\}$ is monotonically nonincreasing. Since $\mathcal{L}$ is bounded below on Θ, the sequence converges to some finite $\mathcal{L}^*$.

Each iterate lies in $\mathcal{S}_0$ by construction, and $\mathcal{S}_0$ is bounded by Lemma 1. Therefore $\{\theta^{(t)}\}$ is bounded and has at least one limit point by the Bolzano–Weierstrass theorem. □

Lemma 3 (Blockwise optimality of limit points). Let $\theta^* = (W^*, H^*, \beta^*)$ be any limit point of $\{\theta^{(t)}\}$ under Assumptions~1–2. Then:

(i) H^* satisfies the KKT conditions for

$$\min_{H \geq 0} \mathcal{L}(W^*, H, \beta^*).$$

(ii) W^* satisfies the KKT conditions for

$$\min_{W \geq 0} \mathcal{L}(W, H^*, \beta^*).$$

(iii) β^* is the unique minimizer of

$$\min_{\beta \in \mathbb{R}^k} \mathcal{L}(W^*, H^*, \beta),$$

and satisfies the KKT conditions for the elastic-net Cox subproblem.

Proof. Let $\{t_j\}$ be a subsequence such that $\theta^{(t_j)} \to \theta^* = (W^*, H^*, \beta^*)$. By continuity of $\mathcal{L}$, $\mathcal{L}(\theta^{(t_j)}) \to \mathcal{L}^*$.

(i) Conditional on (W, β), the H-subproblem is

$$\min_{H \geq 0} \frac{(1-\alpha)}{np}(\|X - WH\|_F^2 + \frac{\lambda_H}{nk}||H||_F^2) + \text{const in } H,$$

a strictly convex quadratic program. The multiplicative update for H is known to be a descent method whose fixed points coincide with the KKT points of this constrained subproblem [@seung2001algorithms; @lin2007convergence]. Since the full objective $\mathcal{L}(\theta^{(t)})$ converges, the per-iteration decrease in $\mathcal{L}$ due to the H-update must vanish along $\{t_j\}$. If H^* were not KKT for the H-subproblem given (W^*, β^*), there would exist a feasible descent direction for H at (W^*, β^*), implying that for sufficiently large j the H-update would still produce a strict decrease in $\mathcal{L}$, contradicting convergence. Hence H^* satisfies the KKT conditions.

(ii) Conditional on (H, β), the W-subproblem is differentiable and convex in W (sum of a quadratic term and a negative Cox partial log-likelihood in a linear predictor). The hybrid multiplicative step with backtracking can be viewed as a projected gradient-like update onto the nonnegative orthant. Under mild conditions on the search direction and step sizes, any limit point of such a projected gradient scheme is a KKT point for the constrained subproblem; see, for example, [@tseng2001convergence; @grippo2000convergence]. If W^* did not satisfy the KKT conditions, there would be a feasible descent direction at (W^*, H^*, β^*) and a sufficiently small step that reduces $\mathcal{L}$, which the line search would eventually accept. This would contradict the fact that $\mathcal{L}(\theta^{(t_j)}) \to \mathcal{L}^*$. Therefore W^* is KKT for the W-subproblem.

(iii) For fixed (W, H), the β-subproblem is the convex elastic-net penalized Cox objective. The coordinate descent algorithm converges to the unique minimizer. By Assumption 2(iii),$\beta^{(t+1)}$ is taken to be this minimizer at each iteration. Passing to the limit along $\{t_j\}$ shows that β^* is the unique minimizer of the β-block given (W^*, H^*) and hence satisfies its KKT conditions. □

Theorem 1 (Convergence to a stationary point). Under Assumptions 1 and 2, the sequence $\{\theta^{(t)}\}$ produced by Algorithm~S1 (without W normalization) satisfies:

(a) The objective values $\{\mathcal{L}(\theta^{(t)})\}$ are monotonically nonincreasing and converge to a finite limit $\mathcal{L}^*$.

(b) Every limit point $\theta^* = (W^*, H^*, \beta^*)$ of $\{\theta^{(t)}\}$ is a stationary point of $\mathcal{L}$ on Θ, i.e. it satisfies the first-order KKT conditions for

$$\min_{\theta \in \Theta} \mathcal{L}(\theta).$$

Proof. Part (a) is Lemma 2. For part (b), Lemma 3 shows that any limit point θ^* is blockwise optimal: each block is optimal (in the KKT sense) when the others are held fixed.

The DeSurv loss can be written as

$$\mathcal{L}(\theta) = f(W, H, \beta) + g(\beta) + I_{\{W \geq 0\}}(W) + I_{\{H \geq 0\}}(H),$$

where f is continuously differentiable, $g(\beta) = \lambda\xi\|\beta\|_1$ is a separable convex (possibly nondifferentiable) penalty, and I_C denotes the indicator of a closed convex set C. This is the smooth+nonsmooth composite form considered in block coordinate descent analyses such as [@tseng2001convergence]. In this setting, any point that is optimal with respect to each block individually (a block coordinatewise minimizer) is a stationary point for the full problem: equivalently,

$$0 \in \nabla_{W,H} f(W^*, H^*, \beta^*) + \partial I_{\{W \geq 0\}}(W^*) + \partial I_{\{H \geq 0\}}(H^*),$$

$$0 \in \nabla_\beta f(W^*, H^*, \beta^*) + \partial g(\beta^*),$$

which are precisely the KKT conditions for $\min_{\theta \in \Theta} \mathcal{L}(\theta)$. Hence every limit point of Algorithm~S1 is stationary. □

Effect of column normalization of W

The above analysis omits the column-normalization step for W used in Algorithm~S2. We briefly argue that this normalization does not affect stationarity of limit points.

Let D be a diagonal matrix with strictly positive diagonal entries. The transformation

$$(W, H, \beta) \mapsto (W', H', \beta') = (WD^{-1},\, DH,\, D\beta)$$

preserves both the product WH and the linear predictor $W\beta$, and hence leaves $\mathcal{L}$ invariant. The column-normalization step is exactly such a transformation, with D chosen from the column norms of W.

Let $\{\theta^{(t)}\}$ be the sequence generated by the algorithm without normalization, and let $\{\tilde{\theta}^{(t)}\}$ be the sequence with normalization. For each t there exists a diagonal $D^{(t)}$ with strictly positive entries such that

$$\tilde{\theta}^{(t)} = (W^{(t)} D^{(t),-1},\, D^{(t)} H^{(t)},\, D^{(t)} \beta^{(t)}),$$

and $\mathcal{L}(\tilde{\theta}^{(t)}) = \mathcal{L}(\theta^{(t)})$. Thus limit points of $\{\tilde{\theta}^{(t)}\}$ are obtained from limit points of $\{\theta^{(t)}\}$ by such invertible diagonal scalings.

Since the KKT conditions are expressed in terms of the gradients with respect to WH and $W\beta$, and these quantities are invariant under the above scaling, stationarity is preserved under the transformation. Therefore Theorem~1 implies that every limit point of the *normalized* algorithm is also a stationary point of $\mathcal{L}$ (up to this scaling equivalence), which establishes convergence of the implementation used in practice.

Remark (Coxnet implementation)

Our implementation updates the β block using a Coxnet-style coordinate descent that relies on an approximate Hessian. Consequently, the formal optimality proof is established for an idealized version of the β update that assumes exact second-order information. Nonetheless, in empirical applications the approximate update is numerically stable and yields a monotone decrease in the objective, consistent with the behavior predicted by the idealized analysis.

Supervision on W versus H

Classical nonnegative matrix factorization (NMF) is non-identifiable: for any invertible matrix R with nonnegative entries, the factorization (W, H) can be transformed to $(WR, R^{-1}H)$ without changing the reconstruction error, yielding multiple valid solutions [@donoho2004nmf; @laurberg2008uniqueness; @gillis2014nmf]. Although normalizing columns of W or rows of H resolves the trivial scaling ambiguity, nonnegative mixing transformations persist, and empirical studies consistently show that the sample-loading matrix H is more sensitive to initialization and local minima than the program matrix W [@brunet2004metagenes; @kim2007sparse; @gillis2012accelerated].

In DeSurv, survival supervision enters through the projection $Z = W^\top X$ in the partial log-likelihood, so that the gradient of the DeSurv loss with respect to the programs is

$$\nabla\mathcal{L} = (1-\alpha)\nabla_W \mathcal{L}_{\text{Cox}}(W, H) - \alpha\nabla_W \mathcal{L}_{\text{Cox}}(W, \beta),$$

and therefore the update rule for W as in Equation~6 relies on both the reconstruction term and the supervision term. As such, the gene-program definitions are directly shaped by their association with survival. This mechanism amplifies genes aligned with the hazard gradient and suppresses those that dilute prognostic structure, ensuring that the learned programs encode survival-relevant biological variation.

By contrast, if supervision were applied to the sample loadings H, the model could reduce the Cox loss by redistributing patient-specific coefficients while leaving the gene-level programs W nearly unchanged, a behavior documented in multiple supervised variants of NMF and supervised topic models, where supervision applied only to the coefficient matrix modifies H but not the basis W [@cai2011graph; @wang2014supervised; @blei2007slda].

Supervising W also provides a practical advantage for **portability**: because W defines gene-level programs, new samples can be embedded by the closed-form projection $Z_{\text{new}} = X_{\text{new}}^\top W$. In contrast, supervising H would not yield such a mapping; instead, obtaining sample loadings for external datasets would require solving a nonnegative least-squares (NNLS) problem for each new sample, introducing optimization noise and reducing reproducibility.

Together, these considerations motivate supervising through W, which shapes the gene programs toward prognostic directions, stabilizes solutions across initializations, and yields biologically interpretable signatures that generalize across cohorts.

Cross-validation procedure

Algorithm~S3 describes the cross validation procedure with F folds and R initializations. We define the c-index using comparable pairs $\mathcal{P} = \{(i,j) : y_i < y_j,\ \delta_i = 1\}$ and linear predictors $\hat{\eta}_i$:

$$\hat{c} = \frac{1}{|\mathcal{P}|}\sum_{(i,j)\in\mathcal{P}} \left[\nVdash(\hat{\eta}_i > \hat{\eta}_j) + \tfrac{1}{2}\nVdash(\hat{\eta}_i = \hat{\eta}_j)\right]. \tag{16}$$

Algorithm 3 Cross-validation for DeSurv pipeline

```
Input:
    Number of folds $F$
    Number of initializations $R$
    Observed data $(X, y, \delta)$
    Hyperparameters $k$, $\alpha$, $\lambda$, $\xi$, and $\lambda_H$
Output: The cross-validated c-index
Divide subjects into $F$ folds.
for $f = 1$ to $F$ do
    Split data into training and validation $(X_{(-f)}, y_{(-f)}, \delta_{(-f)})$, and $(X_{(f)}, y_{(f)}, \delta_{(f)})$ sets
    for $r = 1$ to $R$ do
        set.seed($r$)
        Apply Algorithm 1 with inputs $(X_{(-f)}, y_{(-f)}, \delta_{(-f)})$, and $(k, \alpha, \lambda, \xi, \lambda_H)$
        Obtain $\hat{W}$ and $\hat{\beta}$ as output from Algorithm 1
        Compute the estimated linear predictor: $\hat{\eta} = X_{(f)}^T \hat{W} \hat{\beta}$
        Compute c-index, denoted $\hat{c}_{(f)r}$, according to Equation 16
    end for
    end loop over $r$
    Compute average c-index across initializations: $\hat{c}_{(f)} = \frac{1}{r} \sum_{r=1}^{R} \hat{c}_{(f)r}$
end for
end loop over $f$
Compute final cross-validated c-index: $\hat{c} = \frac{1}{F} \sum_{f=1}^{F} \hat{c}_{(f)}$
return Final estimate $\hat{c}$
```

Bayesian Optimization

We treat hyperparameter tuning as a black-box optimization problem over a bounded domain. Let θ collect the tuned parameters (at minimum $\theta = (k, \alpha, \lambda, \xi)$, and optionally discrete choices such as the signature size n_{top}). For a given θ we run the cross-validation procedure in Algorithm~S3, averaging the c-index across folds and random initializations to form the objective

$$\hat{c}(\theta) = \frac{1}{F} \sum_{f=1}^{F} \hat{c}_{(f)}(\theta).$$

Because $\hat{c}(\theta)$ is expensive to evaluate and non-differentiable with respect to θ, we use Bayesian optimization (BO) to adaptively select candidate configurations.

We place a Gaussian-process (GP) prior on the response surface $f(\theta) \sim \mathcal{GP}(0, k(\theta, \theta'))$ with a Matern kernel and automatic relevance determination length-scales. Continuous parameters are rescaled to $[0, 1]$ and penalties are optimized on a log scale; integer-valued coordinates (e.g., k or n_{top}) are proposed in the continuous space and rounded inside the evaluator. After an initial batch of evaluations, the GP is refit after each new observation and used to guide acquisition.

At iteration t, let $\mu_t(\theta)$ and $\sigma_t(\theta)$ denote the GP posterior mean and standard deviation, and let $f_\star$ be the best observed score. We select new candidates by maximizing expected improvement (EI),

$$\text{EI}(\theta) = (\mu_t(\theta) - f_\star - \kappa)\, \Phi(Z(\theta)) + \sigma_t(\theta)\phi(Z(\theta)), \quad Z(\theta) = \frac{\mu_t(\theta) - f_\star - \kappa}{\sigma_t(\theta)},$$

where Φ and ϕ are the standard normal cdf/pdf and $\kappa \geq 0$ controls the exploration margin. EI is evaluated over a candidate pool sampled from the bounded search region, and the maximizer is evaluated next. The loop continues until the iteration budget is exhausted or the improvement plateaus.

For model selection, we take the highest observed $\hat{c}(\theta)$ as the optimal configuration. When fold-level standard errors are available, we apply a one-standard-error rule for k, selecting the smallest rank whose mean performance lies within one standard error of the best. The final DeSurv model is refit at the selected parameters using the consensus-based initialization described above, and the BO-chosen n_{top} (when tuned) is used to truncate each program for downstream analysis.

PDAC Datasets

We analyzed PDAC cohorts from TCGA-PAAD and CPTAC-PDAC as training data [@raphael2017integrated;@ellis2013clinical]. External validation used independent cohorts from Dijk, Moffitt, PACA-AU (array and RNA-seq), and Puleo [@dijk2020unsupervised;@moffitt2015virtual;@zhao2018gene;@puleo2018stratification]. Training models were fit on the combined TCGA and CPTAC expression matrices after harmonizing gene identifiers; validation datasets were processed with the same gene set and transformations before prediction and clustering.

Dataset-specific inclusion and preprocessing steps were as follows. For TCGA-PAAD, we excluded samples lacking tumor grade and mapped features to gene symbols. For CPTAC, we retained samples annotated as PDAC by histology. For Moffitt, only primary tumor specimens were kept. For PACA-AU array and RNA-seq, we retained primary PDAC tumors and collapsed duplicated gene symbols by median; the RNA-seq cohort received a “_seq’ ’ suffix on sample IDs to avoid collisions with the array cohort. Puleo required no additional dataset-specific filtering beyond survival QC.

Across cohorts, samples without valid survival outcomes (nonpositive time, missing time, or missing event indicator) were removed. Gene filtering and normalization followed the training configuration (ngene and within-sample rank transformation) to mitigate platform effects; validation data were restricted to the training gene set and transformed in the same way.

Bladder Datasets

Bladder analyses used the IMVigor210 cohort with overall survival and clinical annotations [@mariathasan2018tgfbeta]. We performed an internal train/validation split (80/20), stratified by event status, and used the held-out split for evaluation. Sample identifiers and gene features were aligned to the provided gene symbols, and consensus molecular subtype labels were used to exclude NE-like tumors prior to model fitting. Samples with invalid survival outcomes were removed. Gene filtering and normalization followed the training configuration, and the validation split was restricted to the same gene set and transformation used during training.

Survival Analysis

Survival analyses used overall survival time with event indicators as provided by each cohort. For fitted DeSurv models, sample-level program scores were computed as $Z = \tilde{W}^{\top} X$ and standardized using the training mean and standard deviation before constructing the linear predictor $\hat{\eta} = Z\hat{\beta}$. Prognostic performance was summarized using the concordance index as defined above. For visualization and group-level testing, we generated Kaplan–Meier curves and log-rank tests for risk groups defined by median splits of $\hat{\eta}$ within each dataset to avoid cross-cohort scaling differences. When multiple datasets were pooled, Cox models were stratified by dataset to allow cohort-specific baseline hazards. All p-values reported for survival differences between groups correspond to two-sided log-rank tests.