PNAS
www.pnas.org

Supplementary Information for

Conformal Prediction for Biological Design

Clara Fannjiang, Stephen Bates, Anastasios N. Angelopoulos, Jennifer Listgarten, and Michael I. Jordan

Clara Fannjiang and Michael I. Jordan.
E-mails: clarafy@berkeley.edu, jordan@cs.berkeley.com

This PDF file includes:

Supplementary text
Figs. S1 to S6 (not allowed for Brief Reports)
SI References

Supporting Information Text

S1. Proofs

A. Proof of Theorem 1.

Data from feedback covariate shift (FCS) are a special case of what we call *pseudo-exchangeable*[*] random variables.

Definition S1. *Random variables $V_1, \dots, V_{n+1}$ are* pseudo-exchangeable *with factor functions $g_1, \dots, g_{n+1}$ and core function h if the density, f, of their joint distribution can be factorized as*

$$f(v_1, \dots, v_{n+1}) = \prod_{i=1}^{n+1} g_i(v_i;\, v_{-i}) \cdot h(v_1, \dots, v_{n+1}),$$

where $v_{-i} = v_{1:(n+1)} \setminus v_i$,[†] *each $g_i(\cdot;\, v_{-i})$ is a function that depends on the multiset v_{-i} (that is, on the values in v_{-i} but not on their ordering), and h is a function that does not depend on the ordering of its $n+1$ inputs.*

The following lemma characterizes the distribution of the scores of pseudo-exchangeable random variables, which allows for a pseudo-exchangeable generalization of conformal prediction in Theorem S1. We then show that data generated under FCS are pseudo-exchangeable, and a straightforward application of Theorem S1 yields Theorem 1 as a corollary. Our technical development here builds upon the work of Tibshirani et al. (1), who generalized conformal prediction to handle "weighted exchangeable" random variables, including data under standard covariate shift.

The key insight is that if we condition on the values, but not the ordering, of the scores, we can exactly describe their distribution. The following proposition is a generalization of arguments found in the proof of Lemma 3 in (1); the subsequent result in Lemma 1 is a generalization of that lemma.

Proposition 1. *Let $Z_1, \dots, Z_{n+1}$ be pseudo-exchangeable random variables with a joint density function, f, that can be written with factor functions $g_1, \dots, g_{n+1}$ and core function h. Let S be any score function and denote $S_i = S(Z_i, Z_{-i})$ where $Z_{-i} = Z_{1:(n+1)} \setminus \{Z_i\}$ for $i = 1, \dots, n+1$. Define*

$$w_i(z_1, \dots, z_{n+1}) \equiv \frac{\sum_{\sigma : \sigma(n+1) = i} \prod_{j=1}^{n+1} g_j(z_{\sigma(j)}; z_{-\sigma(j)})}{\sum_{\sigma} \prod_{j=1}^{n+1} g_j(z_{\sigma(j)}; z_{-\sigma(j)})}, \quad i = 1, \dots, n+1, \tag{S1}$$

where the summations are taken over permutations, σ, of the integers $1, \dots, n+1$. For values $z = (z_1, \dots, z_{n+1})$, let $s_i = S(z_i, z_{-i})$ and let E_z be the event that $\{Z_1, \dots, Z_{n+1}\} = \{z_1, \dots, z_{n+1}\}$ (that is, the multiset of values taken on by $Z_1, \dots, Z_{n+1}$ equals the multiset of the values in z). Then

$$S_{n+1} \mid E_z \sim \sum_{i=1}^{n+1} w_i(z_1, \dots, z_{n+1})\, \delta_{s_i}.$$

Proof. For simplicity, we treat the case where $S_1, \dots, S_{n+1}$ are distinct almost surely; the result also holds in the general case, but the notation that accommodates duplicate values is cumbersome. For $i = 1, \dots, n+1$,

$$\begin{aligned}
\mathbb{P}(S_{n+1} = s_i \mid E_z) = \mathbb{P}(Z_{n+1} = z_i \mid E_z) &= \frac{\sum_{\sigma : \sigma(n+1) = i} f(z_{\sigma(1)}, \dots, z_{\sigma(n+1)})}{\sum_{\sigma} f(z_{\sigma(1)}, \dots, z_{\sigma(n+1)})} \\
&= \frac{\sum_{\sigma : \sigma(n+1) = i} \prod_{j=1}^{n+1} g_j(z_{\sigma(j)}; z_{-\sigma(j)}) \cdot h(z_{\sigma(1)}, \dots, z_{\sigma(n+1)})}{\sum_{\sigma} \prod_{j=1}^{n+1} g_j(z_{\sigma(j)}; z_{-\sigma(j)}) \cdot h(z_{\sigma(1)}, \dots, z_{\sigma(n+1)})} \\
&= \frac{\sum_{\sigma : \sigma(n+1) = i} \prod_{j=1}^{n+1} g_j(z_{\sigma(j)}; z_{-\sigma(j)}) \cdot h(z_1, \dots, z_{n+1})}{\sum_{\sigma} \prod_{j=1}^{n+1} g_j(z_{\sigma(j)}; z_{-\sigma(j)}) \cdot h(z_1, \dots, z_{n+1})} \\
&= \frac{\sum_{\sigma : \sigma(n+1) = i} \prod_{j=1}^{n+1} g_j(z_{\sigma(j)}; z_{-\sigma(j)})}{\sum_{\sigma} \prod_{j=1}^{n+1} g_j(z_{\sigma(j)}; z_{-\sigma(j)})} \\
&= w_i(z_1, \dots, z_{n+1}).
\end{aligned}$$

□

[*] The name *pseudo-exchangeable* hearkens to the similarity of the factorized form to the pseudo-likelihood approximation of a joint density. Note, however, that each factor, $g_i(v_i;\ v_{-i})$, can only depend on the values and not the ordering of the other variables, $v_1, \dots, v_{i-1}, v_{i+1}, \dots, v_n$, whereas each factor in the pseudo-likelihood approximation also depends on the identities (i.e., the ordering) of the other variables.

[†] With some abuse of notation, we denote $z_{-i} = z_{1:(n+1)} \setminus z_i$ whenever possible, as done here, but use $z_{-i} = z_{1:n} \setminus z_i$ whenever we need to append a candidate test point, as done in the main text and in Theorem S1 below. In either case, we will clarify.

Lemma 1. *Let $Z_1, \dots, Z_{n+1}$ be pseudo-exchangeable random variables with a joint density function, f, that can be written with factor functions $g_1, \dots, g_{n+1}$ and core function h. Let S be any score function and denote $S_i = S(Z_i, Z_{-i})$ where $Z_{-i} = Z_{1:(n+1)} \setminus \{Z_i\}$ for $i = 1, \dots, n+1$. For any $\beta \in (0,1)$,*

$$\mathbb{P}\left\{ S_{n+1} \leq \textsc{Quantile}_\beta \left(\sum_{i=1}^{n+1} w_i(Z_1, \dots, Z_{n+1}) \, \delta_{S_i} \right) \right\} \geq \beta,$$

where $w_i(z_1, \dots, z_{n+1})$ is defined in Eq. [S1].

Proof. Assume for simplicity of notation that $S_1, \dots, S_{n+1}$ are distinct almost surely (but the result holds generally). For data point values $z = (z_1, \dots, z_{n+1})$, let $s_i = S(z_i, z_{-i})$ and let E_z be the event that $\{Z_1, \dots, Z_{n+1}\} = \{z_1, \dots, z_{n+1}\}$. By Proposition 1,

$$S_{n+1} \mid E_z \sim \sum_{i=1}^{n+1} w_i(z_1, \dots, z_{n+1}) \, \delta_{s_i},$$

and consequently

$$\mathbb{P}\left(S_{n+1} \leq \textsc{Quantile}_\beta \left(\sum_{i=1}^{n+1} w_i(z_1, \dots, z_{n+1}) \, \delta_{s_i} \right) \middle| \, E_z \right) \geq \beta,$$

by definition of the β-quantile; equivalently, since we condition on E_z,

$$\mathbb{P}\left(S_{n+1} \leq \textsc{Quantile}_\beta \left(\sum_{i=1}^{n+1} w_i(Z_1, \dots, Z_{n+1}) \, \delta_{S_i} \right) \middle| \, E_z \right) \geq \beta.$$

Since this inequality holds for all events E_z, where z is a vector of $n+1$ data point values, smoothing gives

$$\mathbb{P}\left(S_{n+1} \leq \textsc{Quantile}_\beta \left(\sum_{i=1}^{n+1} w_i(Z_1, \dots, Z_{n+1}) \, \delta_{S_i} \right) \right) \geq \beta.$$

□

Lemma 1 yields the following theorem, which enables a generalization of conformal prediction to pseudo-exchangeable random variables.

Theorem S1. *Suppose $Z_1, \dots, Z_{n+1}$ where $Z_i = (X_i, Y_i) \in \mathcal{X} \times \mathbb{R}$ are pseudo-exchangeable random variables with factor functions $g_1, \dots, g_{n+1}$. For any score function, S, and any miscoverage level, $\alpha \in (0,1)$, define for any point $x \in \mathcal{X}$:*

$$C_\alpha(x) = \left\{ y \in \mathbb{R} : S_{n+1}(x,y) \leq \textsc{Quantile}_{1-\alpha} \left(\sum_{i=1}^{n+1} w_i(Z_1, \dots, Z_n, (x,y)) \, \delta_{S_i(x,y)} \right) \right\}, \qquad \text{[S2]}$$

where $S_i(x,y) = S(Z_i, Z_{-i} \cup \{(x,y)\})$ and $Z_{-i} = Z_{1:n} \setminus Z_i$ for $i = 1, \dots, n$, $S_{n+1}(x,y) = S((x,y), Z_{1:n})$, and the weight functions w_i are as defined in Eq. [S1]. Then C_α satisfies

$$\mathbb{P}\left(Y_{n+1} \in C_\alpha(X_{n+1}) \right) \geq 1 - \alpha,$$

where the probability is over all $n+1$ data points, $Z_1, \dots, Z_{n+1}$.

Proof. By construction, we have

$$Y_{n+1} \in C_\alpha(X_{n+1}) \iff S_{n+1}(X_{n+1}, Y_{n+1}) \leq \textsc{Quantile}_{1-\alpha} \left(\sum_{i=1}^{n+1} w_i(Z_1, \dots, Z_{n+1}) \, \delta_{S_i(X_{n+1}, Y_{n+1})} \right).$$

Applying Lemma 1 gives the result. □

Finally, Theorem 1 follows as a corollary of Theorem S1. Denoting $Z_{n+1} = Z_{\text{test}}$ and $Z_{-i} = Z_{1:(n+1)} \setminus \{Z_i\}$, observe that data, $(Z_1, \dots, Z_{n+1})$, under FCS are pseudo-exchangeable with the core function

$$h(z_1, \dots, z_{n+1}) = \prod_{i=1}^{n+1} p_X(x_i) \, p_{Y|X}(y_i \mid x_i),$$

and factor functions $g_i(z_i;\, z_{-i}) = 1$ for $i = 1, \dots, n$ and

$$g_{n+1}(z_{n+1};\, z_{1:n}) = \frac{\tilde{p}_{X;z_{1:n}}(x_{n+1})\, p_{Y|X}(y_{n+1} \mid x_{n+1})}{p_X(x_{n+1})\, p_{Y|X}(y_{n+1} \mid x_{n+1})} = \frac{\tilde{p}_{X;z_{1:n}}(x_{n+1})}{p_X(x_{n+1})} = v(x_{n+1}; z_{1:n})$$

where $v(\cdot;\cdot)$ is the likelihood ratio function defined in Eq. [2]. The weights, $w_i(z_1, \dots, z_{n+1})$, in Eq. [S1] then simplify as

$$\begin{aligned} w_i(z_1, \dots, z_{n+1}) = \frac{\sum_{\sigma:\sigma(n+1)=i} \prod_{j=1}^{n+1} g_j(z_{\sigma(j)}; z_{-\sigma(j)})}{\sum_{\sigma} \prod_{j=1}^{n+1} g_j(z_{\sigma(j)}; z_{-\sigma(j)})} &= \frac{\sum_{\sigma:\sigma(n+1)=i} \prod_{j=1}^{n+1} g_j(z_{\sigma(j)}; z_{-\sigma(j)})}{\sum_{k=1}^{n+1} \sum_{\sigma:\sigma(n+1)=k} \prod_{j=1}^{n+1} g_j(z_{\sigma(j)}; z_{-\sigma(j)})} \\ &= \frac{\sum_{\sigma:\sigma(n+1)=i} g_{n+1}(z_{\sigma(n+1)}; z_{-\sigma(n+1)})}{\sum_{k=1}^{n+1} \sum_{\sigma:\sigma(n+1)=k} g_{n+1}(z_{\sigma(n+1)}; z_{-\sigma(n+1)})} \\ &= \frac{\sum_{\sigma:\sigma(n+1)=i} g_{n+1}(z_i; z_{-i})}{\sum_{k=1}^{n+1} \sum_{\sigma:\sigma(n+1)=k} g_{n+1}(z_k; z_{-k})} \\ &= \frac{n! \cdot g_{n+1}(z_i; z_{-i})}{\sum_{k=1}^{n+1} n! \cdot g_{n+1}(z_k; z_{-k})} \\ &= \frac{v(x_i; z_{-i})}{\sum_{k=1}^{n+1} v(x_k; z_{-k})}. \end{aligned}$$

These quantities are exactly the weight functions, w_i^y, defined in Eq. [4] and used in the full conformal confidence set in Eq. [3]: $w_i^y(X_{\text{test}}) = w_i(Z_1, \dots, Z_n, (X_{\text{test}}, y))$ for $i = 1, \dots, n+1$. That is, Eq. [3] gives the confidence set defined in Eq. [S2] for data under FCS. Applying Theorem S1 then yields Theorem 1.

B. A randomized confidence set achieves exact coverage. Here, we introduce the *randomized β-quantile* and a corresponding randomized confidence set that achieves exact coverage. To lighten notation, for a discrete distribution with probability masses $w = (w_1, \dots, w_{n+1})$ on points $s = (s_1, \dots, s_{n+1})$, where $s_i \in \mathbb{R}$ and $w_i \geq 0, \sum_{i=1}^{n+1} w_i = 1$, we will write $\text{QUANTILE}_\beta(s, w) = \text{QUANTILE}_\beta(\sum_{i=1}^{n} w_i \delta_{s_i})$. Observe that $\text{QUANTILE}_\beta(s, w)$ is always one of the support points, s_i. Now define the *β-quantile lower bound*:

$$\text{QUANTILELB}_\beta\,(s, w) = \inf\left\{ s : \sum_{i:s_i \leq s} w_i < \beta, \sum_{i:s_i \leq s} w_i + \sum_{j:s_j = \text{QUANTILE}_\beta(s,w)} w_j \geq \beta \right\},$$

which is either a support point strictly less than the β-quantile, or negative infinity. Finally, letting $\text{QF}_\beta(s, w)$ and $\text{LF}_\beta(s, w)$ denote the CDF of the discrete distribution at $\text{QUANTILE}_\beta(s, w)$ and $\text{QUANTILELB}_\beta(s, w)$), respectively, the randomized β-quantile is a random variable that takes on the value of either the β-quantile or the β-quantile lower bound:

$$\text{RANDOMIZEDQUANTILE}_\beta(s, w) = \begin{cases} \text{QUANTILELB}_\beta(s, w) & \text{w. p. } \frac{\text{QF}_\beta(s,w) - \beta}{\text{QF}_\beta(s,w) - \text{LF}_\beta(s,w)}, \\ \text{QUANTILE}_\beta(s, w) & \text{w. p. } 1 - \frac{\text{QF}_\beta(s,w) - \beta}{\text{QF}_\beta(s,w) - \text{LF}_\beta(s,w)}. \end{cases} \qquad [S3]$$

We use this quantity to define the *randomized full conformal* confidence set, which, for any miscoverage level, $\alpha \in (0, 1)$, and $x \in \mathcal{X}$ is the following random variable:

$$C_\alpha^{\text{rand}}(x) = \left\{ y \in \mathbb{R} : S((x, y), Z_{1:n}) \leq \text{RANDOMIZEDQUANTILE}_{1-\alpha}(s(Z_1, \dots, Z_n, (x, y)), w(Z_1, \dots, Z_n, (x, y)) \right\}, \qquad [S4]$$

where $s(Z_1, \dots, Z_n, (x, y)) = (S_1, \dots, S_n, S((x, y), Z_{1:n})$ and $S_i = S(Z_i, Z_{-i} \cup \{(x, y)\})$ for $i = 1, \dots, n$, and $w(Z_1, \dots, Z_n, (x, y)) = (w_1^y(x), \dots, w_{n+1}^y(x))$ where $w_i^y(x)$ is defined in Eq. [4]. Note that for each candidate label, $y \in \mathbb{R}$, an independent randomized β-quantile is instantiated; some values will use the β-quantile as the threshold on the score, while the others will use the β-quantile lower bound. Randomizing the confidence set in this way yields the following result.

Theorem S2. *Suppose data, $Z_1, \dots, Z_n, Z_{\text{test}}$, are generated under feedback covariate shift and assume $\tilde{P}_{X;D}$ is absolutely continuous with respect to P_X for all possible values of D. Then, for any miscoverage level, $\alpha \in (0, 1)$, the randomized full confidence set, C_α^{rand}, in Eq. [S4] satisfies the* exact coverage *property:*

$$\mathbb{P}(Y_{\text{test}} \in C_\alpha^{\text{rand}}(X_{\text{test}})) = 1 - \alpha, \qquad [S5]$$

where the probability is over $Z_1, \dots, Z_n, Z_{\text{test}}$ and the randomness in C_α^{rand}.

Proof. Denote $Z_{n+1} = Z_{\text{test}}$ and $Z = (Z_1, \dots, Z_{n+1})$. For a vector of $n+1$ data point values, $z = (z_1, \dots, z_{n+1})$, use the following shorthand:

$$\begin{aligned} Q_\beta(z) &= \textsc{Quantile}_\beta(s(z), w(z)), \\ L_\beta(z) &= \textsc{QuantileLB}_\beta(s(z), w(z)), \\ R_\beta(z) &= \textsc{RandomizedQuantile}_\beta(s(z), w(z)), \\ \mathrm{QF}_\beta(z) &= \mathrm{QF}_\beta(s(z), w(z)), \\ \mathrm{LF}_\beta(z) &= \mathrm{LF}_\beta(s(z), w(z)). \end{aligned}$$

As in the proof of Lemma 1, consider the event, E_z, that $\{Z_1, \dots, Z_{n+1}\} = \{z_1, \dots, z_{n+1}\}$. Assuming for simplicity that the scores are distinct almost surely, by Proposition 1

$$S(Z_{n+1}, Z_{1:n}) \mid E_z \sim \sum_{i=1}^{n+1} w_i(z_1, \dots, z_{n+1})\, \delta_{S(z_i, z_{-i})},$$

and consequently

$$\begin{aligned} &\mathbb{P}(S(Z_{n+1}, Z_{1:n}) \le \mathrm{R}_{1-\alpha}(z) \mid E_z) \\ &= \mathbb{P}(S(Z_{n+1}, Z_{1:n}) \le \mathrm{R}_{1-\alpha}(z) \mid E_z, \mathrm{R}_{1-\alpha}(z) = \mathrm{Q}_{1-\alpha}(z)) \cdot \mathbb{P}(\mathrm{R}_{1-\alpha}(z) = \mathrm{Q}_{1-\alpha}(z) \mid E_z) + \\ &\qquad \mathbb{P}(S(Z_{n+1}, Z_{1:n}) \le \mathrm{R}_{1-\alpha}(z) \mid E_z, \mathrm{R}_{1-\alpha}(z) = \mathrm{L}_{1-\alpha}(z)) \cdot \mathbb{P}(\mathrm{R}_{1-\alpha}(z) = \mathrm{L}_{1-\alpha}(z) \mid E_z) \\ &= \mathbb{P}(S(Z_{n+1}, Z_{1:n}) \le \mathrm{Q}_{1-\alpha}(z) \mid E_z) \cdot \left(1 - \frac{\mathrm{QF}_{1-\alpha}(z) - (1-\alpha)}{\mathrm{QF}_{1-\alpha}(z) - \mathrm{LF}_{1-\alpha}(z)}\right) + \\ &\qquad \mathbb{P}(S(Z_{n+1}, Z_{1:n}) \le \mathrm{L}_{1-\alpha}(z) \mid E_z) \cdot \frac{\mathrm{QF}_{1-\alpha}(z) - (1-\alpha)}{\mathrm{QF}_{1-\alpha}(z) - \mathrm{LF}_{1-\alpha}(z)} \\ &= \mathrm{QF}_{1-\alpha}(z) \cdot \left(1 - \frac{\mathrm{QF}_{1-\alpha}(z) - (1-\alpha)}{\mathrm{QF}_{1-\alpha}(z) - \mathrm{LF}_{1-\alpha}(z)}\right) + \mathrm{LF}_{1-\alpha}(z) \cdot \frac{\mathrm{QF}_{1-\alpha}(z) - (1-\alpha)}{\mathrm{QF}_{1-\alpha}(z) - \mathrm{LF}_{1-\alpha}(z)} \\ &= -\left(\mathrm{QF}_{1-\alpha}(z) - \mathrm{LF}_{1-\alpha}(z)\right) \cdot \frac{\mathrm{QF}_{1-\alpha}(z) - (1-\alpha)}{\mathrm{QF}_{1-\alpha}(z) - \mathrm{LF}_{1-\alpha}(z)} + \mathrm{QF}_{1-\alpha}(z) \\ &= -\mathrm{QF}_{1-\alpha}(z) + (1-\alpha) + \mathrm{QF}_{1-\alpha}(z) \\ &= 1 - \alpha. \end{aligned}$$

Since we condition on E_z, we equivalently have

$$\mathbb{P}(S(Z_{n+1}, Z_{1:n}) \le R_{1-\alpha}(Z) \mid E_z) = 1 - \alpha,$$

and since this equality holds for all events E_z, where z is a vector of $n+1$ data point values, taking an expectation over E_z yields

$$\mathbb{P}(S(Z_{n+1}, Z_{1:n}) \le R_{1-\alpha}(Z)) = 1 - \alpha.$$

Finally, since

$$Y_{n+1} \in C_\alpha^{\text{rand}}(X_{n+1}) \iff S(Z_{n+1}, Z_{1:n}) \le R_{1-\alpha}(Z),$$

the result follows. □

Note that standard covariate shift is subsumed by feedback covariate shift, so Theorem S2 can be used to construct a randomized confidence set with exact coverage under standard covariate shift as well.

C. Data splitting. In general, computing the full conformal confidence set, $C_\alpha(x)$, using Alg. 1 requires fitting $(n+1) \times |\mathcal{Y}|$ regression models. A much more computationally attractive alternative is called a *data splitting* or *split conformal* approach (2, 3), in which we (i) randomly partition the labeled data into disjoint training and *calibration* data sets, (ii) fit a regression model to the training data, and (iii) use the scores that it provides for the calibration data (but not the training data) to construct confidence sets for test data points. Though this approach only requires fitting a single model, the trade-off is that it does not use the labeled data as efficiently: only some fraction of our labeled data can be used to train the regression model. This limitation may be inconsequential for settings with abundant data, but can be a nonstarter when labeled data is limited, such as in many protein design problems.

Here, we show how data splitting simplifies feedback covariate shift (FCS) to standard covariate shift. We then use the data splitting method from Tibshirani et al. (1) to produce confidence sets with coverage; the subsequent subsection shows how to introduce randomization to achieve exact coverage.

To begin, we recall the standard covariate shift model (4–6). The training data, $Z_1, \ldots, Z_n$ where $Z_i = (X_i, Y_i)$, are i.i.d. from some distribution: $X_i \sim P_X, Y_i \sim P_{Y \mid X_i}$ for $i = 1, \ldots, n$. A test data point, $Z_{\text{test}} = (X_{\text{test}}, Y_{\text{test}})$, is drawn from a different input distribution but the same conditional distribution, $X_{\text{test}} \sim \tilde{P}_X, Y_{\text{test}} \sim P_{Y \mid X_{\text{test}}}$, independently from the training data. In contrast to FCS, here the test input cannot be chosen in a way that depends on the training data.

Returning to FCS, suppose we randomly partition all our labeled data into disjoint training and calibration data sets. Let μ denote the regression model fit to the training data; we henceforth consider μ as fixed and make no further use of the training data. As such, without loss of generality we will use $Z_1, \ldots, Z_m$ to refer to the calibration data. Now suppose the test input distribution is induced by the trained regression model, μ; we write this as $\tilde{P}_{X;\mu}$. Observe that, conditioned on the training data, we now have a setting where the calibration and test data are drawn from different input distributions but the same conditional distribution, $P_{Y \mid X}$, and are independent of each other. That is, data splitting returns us to standard covariate shift.

To construct valid confidence sets under standard covariate shift, define the following likelihood ratio function:

$$v(x) = \frac{\tilde{p}_{X;\mu}(x)}{p_X(x)}, \qquad \text{[S6]}$$

where p_X and $\tilde{p}_{X;\mu}$ refer to the densities of the training and test input distributions, respectively. We restrict our attention to score functions of the following form (7):

$$S(x, y) = \frac{|y - \mu(x)|}{u(x)}. \qquad \text{[S7]}$$

where u is any heuristic, nonnegative notion of uncertainty; one can also set $u(x) = 1$ to recover the residual score function. Note that, since we condition on the training data and treat the regression model as fixed, the score of a point, (x, y), is no longer also a function of other data points. Finally, for any miscoverage level, $\alpha \in (0, 1)$, and any $x \in \mathcal{X}$, define the *split conformal* confidence set as

$$\begin{aligned} C_\alpha^{\text{split}}(x) &= \mu(x) \pm q \cdot u(x), \\ q &= \textsc{Quantile}_{1-\alpha}\left(\sum_{i=1}^{m} w_i(x)\, \delta_{S_i} + w_{n+1}(x)\, \delta_\infty \right), \end{aligned} \qquad \text{[S8]}$$

where $S_i = S(X_i, Y_i)$ for $i = 1, \ldots, m$ and

$$\begin{aligned} w_i(x) &= \frac{v(X_i)}{\sum_{j=1}^m v(X_j) + v(x)}, \quad i = 1, \ldots, m, \\ w_{m+1}(x) &= \frac{v(x)}{\sum_{j=1}^m v(X_j) + v(x)}. \end{aligned} \qquad \text{[S9]}$$

For data under standard covariate shift, the split conformal confidence set achieves coverage, as first shown in (1).

Theorem S3 (Corollary 1 in (1)). *Suppose calibration and test data, $Z_1, \ldots, Z_m, Z_{\text{test}}$, are under standard covariate shift, and assume $\tilde{P}_{X;\mu}$ is absolutely continuous with respect to P_X. For score functions of the form in Eq. [S7], and any miscoverage level, $\alpha \in (0, 1)$, the split conformal confidence set, $C_\alpha^{\text{split}}(x)$, in Eq. [S8] satisfies the coverage property in Eq. [1].*

To achieve exact coverage, we can introduce randomization, as we discuss next.

D. Data splitting with randomization achieves exact coverage. Here, we stay in the setting and notation of the previous subsection and demonstrate how randomizing the β-quantile enables a data splitting approach to achieve exact coverage. For any score function of the form in Eq. [S7], any miscoverage level, $\alpha \in (0, 1)$, the *randomized split conformal* confidence set is the following random variable for $x \in \mathcal{X}$:

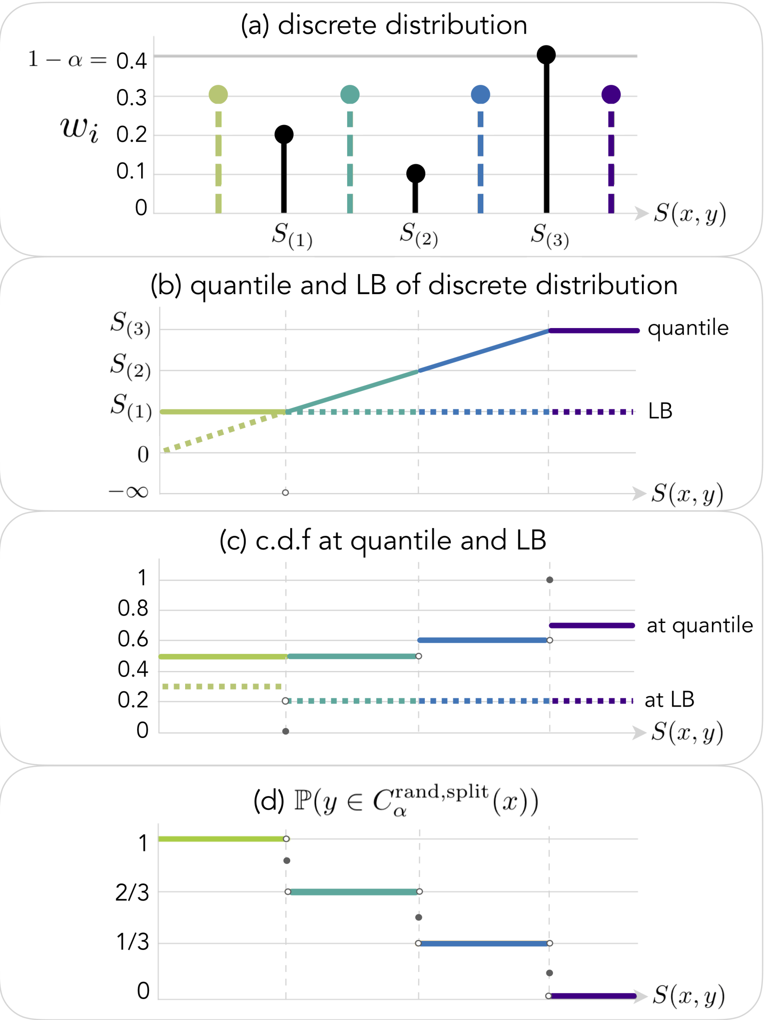
$$C_\alpha^{\text{rand,split}}(x) = \left\{ y \in \mathbb{R} : S(x, y) \leq \textsc{RandomizedQuantile}_{1-\alpha}\left((S_1, \ldots, S_m, S(x, y)), (w_1(x), \ldots, w_{m+1}(x)) \right) \right\}, \qquad \text{[S10]}$$

where the randomized β-quantile, $\textsc{RandomizedQuantile}_\beta$ is defined in Eq. [S3], $S_i = S(X_i, Y_i)$ for $i = 1, \ldots, m$, and $w_i(\cdot)$ for $i = 1, \ldots, m + 1$ is defined in Eq. [S9]. Observe that for each candidate label, $y \in \mathbb{R}$, an independent randomized β-quantile is drawn, such that the scores of some values are compared to the β-quantile while the others are compared to the β-quantile lower bound. The exact coverage property of this confidence set is a consequence of Theorem S2.

Corollary 1. *Suppose calibration and test data, $Z_1, \ldots, Z_m, Z_{\text{test}}$, are under standard covariate shift, and assume $\tilde{P}_{X;\mu}$ is absolutely continuous with respect to P_X. For score functions of the form in Eq. [S7], and any miscoverage level, $\alpha \in (0, 1)$, the randomized split conformal confidence set, $C_\alpha^{\text{rand,split}}(x)$, in Eq. [S10] satisfies the exact coverage property in Eq. [S5].*

Proof. Since standard covariate shift is a special case of FCS, the calibration and test data can be described by FCS where $\tilde{P}_{X;D} = \tilde{P}_{X;\mu}$ for any multiset D. The randomized split conformal confidence set, $C_\alpha^{\text{rand,split}}$, is simply the randomized full conformal confidence set, C_α^{rand}, defined in Eq. [S4], instantiated with the scores $S((x, y), Z_{1:m}) = S(x, y)$ and $S(Z_i, Z_{-i} \cup \{(x, y)\}) = S(Z_i)$ for $i = 1, \ldots, m$, and weights resulting from $\tilde{P}_{X;D} = \tilde{P}_{X;\mu}$ for all D. The result then follows from Theorem S2. □

While we only need to fit a single regression model to compute the scores for data splitting, naively it might seem that in practice, we need to approximate $C_\alpha^{\mathrm{rand,split}}(x)$ by introducing a discrete grid of candidate labels, $\mathcal{Y} \subset \mathbb{R}$, and computing a randomized β-quantile for $|\mathcal{Y}|$ different discrete distributions. Fortunately, we can construct an alternative confidence set that also achieves exact coverage, the *randomized staircase* confidence set, $C_\alpha^{\mathrm{staircase}}$, which only requires sorting m scores and an additional $O(m)$ floating point operations to compute (see Alg. S1).



color legend:

$S(x,y) \in [0, S_{(1)})$	$S(x,y) \in (S_{(1)}, S_{(2)})$	$S(x,y) \in (S_{(2)}, S_{(3)})$	$S(x,y) \in (S_{(3)}, \infty]$
$\textsc{Quantile} = S_{(1)} > S(x,y)$ $\mathbb{P}(y \in C_\alpha^{\mathrm{rand,split}}(x)) = 1$	$\textsc{Quantile} = S(x,y)$ $\textsc{QuantileLB} = S_{(1)}$ $\mathrm{QF} = 0.2 + 0.3$ $\mathrm{LF} = 0.2$ $\mathbb{P}(y \in C_\alpha^{\mathrm{rand,split}}(x)) = 2/3$	$\textsc{Quantile} = S(x,y)$ $\textsc{QuantileLB} = S_{(2)}$ $\mathrm{QF} = 0.2 + 0.1 + 0.3$ $\mathrm{LF} = 0.2 + 0.1$ $\mathbb{P}(y \in C_\alpha^{\mathrm{rand,split}}(x)) = 1/3$	$\textsc{Quantile} = S_{(3)} < S(x,y)$ $\mathbb{P}(y \in C_\alpha^{\mathrm{rand,split}}(x)) = 0$

Fig. S1. Depiction of how the probability $\mathbb{P}(y \in C_\alpha^{\text{rand,split}}(x))$ is a piecewise constant function of y. (a) Given the values of the calibration data and test input, the scores $S_1, \ldots, S_m$ and corresponding probability masses $w_1, \ldots, w_m$ (black stems), as well as the probability mass for the test input, $w_{m+1} = 0.3$, are fixed. The only quantity that depends on y is $S(x,y)$. Four example values are shown as dashed green, teal, blue, and purple stems, representing values in $[0, S_{(1)})$, $(S_{(1)}, S_{(2)})$, $(S_{(2)}, S_{(3)})$, and $(S_{(3)}, \infty]$, respectively (see color legend). Note that in this example, $1-\alpha = 0.4$. (b) The 0.4-quantile and 0.4-quantile lower bound of the discrete distribution in the top panel as a function of $S(x,y)$, where the colors correspond to values of $S(x,y)$ in the intervals just listed. Note the discontinuity in the 0.4-quantile lower bound at $S(x,y) = S_{(1)}$. (c) The c.d.f. of the discrete distribution at the 0.4-quantile and 0.4-quantile lower bound. Note the discontinuities when $S(x,y)$ equals a calibration score. (d) The probability $\mathbb{P}(y \in C_\alpha^{\text{rand,split}}(x))$, which equals 1 or 0 if $S(x,y) = 0.4$-quantile lower bound or $S(x,y) > 0.4$-quantile, respectively, and otherwise equals the probability in Eq. [S3] that the randomized 0.4-quantile equals the 0.4-quantile: $1 - \frac{\text{QF}-0.4}{\text{QF}-\text{LF}}$, where QF and LF denote the c.d.f. at the 0.4-quantile and 0.4-quantile lower bound, respectively. Color legend: calculations of the plotted quantities (calculations for $S(x,y) = S_{(i)}$ omitted).

At a high level, its construction is based on the observation that for any $x \in \mathcal{X}$ and $y \in \mathbb{R}$, the quantity $\mathbb{P}(y \in C_\alpha^{\mathrm{rand,split}}(x))$, where the probability is over the randomness in $C_\alpha^{\mathrm{rand,split}}(x)$, is a piecewise constant function of y. Instead of testing each value of $y \in \mathbb{R}$, we can then construct this piecewise constant function, and randomly include entire intervals of y values that

Algorithm S1 Randomized staircase confidence set

```
Input: Miscoverage level, $\alpha \in (0,1)$; calibration data, $Z_1, \dots, Z_m$, where $Z_i = (X_i, Y_i)$; test input, $X_{\text{test}}$; subroutine for likelihood ratio function, $v(\cdot)$, defined in Eq. [S6]; subroutine for uncertainty heuristic, $u(\cdot)$; subroutine for regression model prediction, $\mu(\cdot)$.
Output: Randomized staircase confidence set, $C = C_\alpha^{\text{staircase}}(X_{\text{test}})$.
for $i = 1, \dots, m$ do                                        ▷ Compute calibration scores
    $S_i \leftarrow |Y_i - \mu(X_i)|/u(X_i)$
    $v_i \leftarrow v(X_i)$
$v_{m+1} \leftarrow v(X_{\text{test}})$
for $i = 1, \dots, m+1$ do                                      ▷ Compute calibration and test weights
    $w_i \leftarrow v_i / \sum_{j=1}^{m+1} v_j$
$C \leftarrow \emptyset$
LowerBoundIsSet $\leftarrow$ False
$S_{(0)} = 0, w_0 = 0$                                          ▷ Dummy values so for-loop will include $[0, S_{(1)}]$
for $i = 0, \dots, m-1$ do
    if $\sum_{j:S_j \le S_{(i)}} w_j + w_{m+1} < 1 - \alpha$ then    ▷ $S(x,y) \le \beta$-quantile lower bound, so include deterministically
        $C = C \cup \Big[\mu(X_{\text{test}}) + S_{(i)} \cdot u(X_{\text{test}}), \mu(X_{\text{test}}) + S_{(i+1)} \cdot u(X_{\text{test}})\Big] \cup \Big[\mu(X_{\text{test}}) - S_{(i+1)} \cdot u(X_{\text{test}}), \mu(X_{\text{test}}) - S_{(i)} \cdot u(X_{\text{test}})\Big]$
    else if $\sum_{j:S_j \le S_{(i)}} w_j + w_{m+1} \ge 1 - \alpha$ and $\sum_{j:S_j \le S_{(i)}} w_j < 1 - \alpha$ then    ▷ $S(x,y) = \beta$-quantile, so randomize inclusion
        if LowerBoundIsSet = False then
            LowerBoundIsSet $\leftarrow$ True                    ▷ Set $\beta$-quantile lower bound
            $LF = \sum_{j:S_j \le S_{(i)}} w_j$
        $F \leftarrow \frac{\sum_{j:S_j \le S_{(i)}} w_j + w_{m+1} - (1-\alpha)}{\sum_{j:S_j \le S_{(i)}} w_j + w_{m+1} - LF}$
        $b \sim \text{Bernoulli}(1 - F)$
        if $b$ then
            $C = C \cup \Big[\mu(X_{\text{test}}) + S_{(i)} \cdot u(X_{\text{test}}), \mu(X_{\text{test}}) + S_{(i+1)} \cdot u(X_{\text{test}})\Big] \cup \Big[\mu(X_{\text{test}}) - S_{(i+1)} \cdot u(X_{\text{test}}), \mu(X_{\text{test}}) - S_{(i)} \cdot u(X_{\text{test}})\Big]$
if $\sum_{i=1}^{m} w_i < 1 - \alpha$ then                       ▷ For $S(x,y) > S_{(m)}$, either $S(x,y) = \beta$-quantile or $S(x,y) > \beta$-quantile
    if LowerBoundIsSet = False then
        $LF = \sum_{i=1}^{m} w_i$
    $F \leftarrow \frac{1-(1-\alpha)}{1-LF}$
    $b \sim \text{Bernoulli}(1 - F)$
    if $b$ then
        $C = C \cup \Big[\mu(X_{\text{test}}) + S_{(m)} \cdot u(X_{\text{test}}), \infty\Big] \cup \Big[-\infty, \mu(X_{\text{test}}) - S_{(m)} \cdot u(X_{\text{test}})\Big]$
```

have the same value of $\mathbb{P}(y \in C_\alpha^{\text{rand,split}}(x))$.

Fig. S1 illustrates this observation, which we now explain. First, the discrete distribution in Eq. [S10] has probability masses $w_1(x), \dots, w_{m+1}(x)$ at the points $S_1, \dots, S_m, S(x,y)$, respectively. Given the values of the m calibration data points and the test input, x, all of these quantities are fixed—except for the score of the candidate test data point, $S(x,y)$. That is, the only quantity that depends on the value of y is $S(x,y)$, which is the location of the probability mass $w_{m+1}(x)$; the remaining m support points and their corresponding probability masses do not not change with y.

Now consider the calibration scores, $S_1, \dots, S_m$, sorted in ascending order. Observe that for any pair of successive sorted scores, $S_{(i)}$ and $S_{(i+1)}$, the entire interval of y values such that $S(x,y) \in (S_{(i)}, S_{(i+1)})$ belongs to one of three categories: $S(x,y) \le \beta$-quantile lower bound (of the discrete distribution with probability masses $w_1, \dots, w_{m+1}$ at support points $S_1, \dots, S_m, S(x,y)$), $S(x,y) = \beta$-quantile, or $S(x,y) > \beta$-quantile. An interval of y values that belongs to the first category is deterministically included in $C_\alpha^{\text{rand,split}}(x)$, regardless of the randomness in the randomized β-quantile (color-coded green in Fig. S1), while an interval that belongs to the last category is deterministically excluded (color-coded purple in Fig. S1). The only y values whose inclusion is not deterministic are those in the second category (color-coded teal and blue), which are randomly included with the probability, given in Eq. [S3], that the randomized β-quantile equals the β-quantile. Consequently, we can identify the intervals of y values belonging to each of these categories, and for those in the second category, compute the probability that the randomized β-quantile is instantiated as the β-quantile, which is $\mathbb{P}(y \in C_\alpha^{\text{rand,split}}(x))$.

This probability turns out to be a piecewise constant function of y. Note that it is computed from two quantities: the c.d.f. at the β-quantile and the c.d.f at the β-quantile lower bound (see Eq. [S3]). As depicted in Fig. S1 (third panel from top), for any two successive sorted calibration scores, $S_{(i)}$ and $S_{(i+1)}$, both of these quantities are constant over $S(x,y) \in (S_{(i)}, S_{(i+1)})$. That is, both the c.d.f. at the β-quantile and the c.d.f. at β-quantile lower bound are piecewise constant functions of y, which only change values at the calibration scores, $S_1, \dots, S_m$ (and can take on different values exactly at the calibration scores). Consequently, the probability $\mathbb{P}(y \in C_\alpha^{\text{rand,split}}(x))$ is also a piecewise constant function of y, which only changes values at the calibration scores. It attains its highest value at $\mu(x)$ and decreases as y moves further away from it, resembling a staircase, as depicted in Fig. S1 (fourth panel from the top).

Therefore, instead of computing a randomized β-quantile for all $y \in \mathbb{R}$, we can simply compute the value of this probability on the $m+1$ intervals between neighboring sorted calibration scores: $[0, S_{(1)}), (S_{(1)}, S_{(2)}), \dots, (S_{(m-1)}, S_{(m)}), (S_{(m)}, \infty]$, as well as its value exactly at the m calibration scores. These probabilities may equal 1 or 0, which correspond to the two cases earlier described wherein y is deterministically included or excluded, respectively. If the probability is not 1 or 0, then we can randomly include the entire set of values of y such that $S(x,y)$ falls in the interval. Due to the form of the score in Eq. [S7], this set comprises two equal-length intervals on both sides of $\mu(x)$: $(\mu(x) - S_{(i+1)}, \mu(x) - S_{(i)}) \cup (\mu(x) + S_{(i+1)}, \mu(x) + S_{(i)})$.

Finally, if we assume that scores are distinct almost surely, then our treatment of values of y such that $S(x, y) = S_i$ for $i = 1, \dots, m$, does not affect the exact coverage property. For simplicity, Alg. S1 therefore includes or excludes closed intervals that contain these y values as endpoints, rather than treating them separately.

More general score functions. In the reasoning above, we use the assumption that the score function takes the form in Eq. [S7] only at the end of the argument, to infer the form of the sets of y values. We can relax this assumption as follows. For any continuous score function, consider the preimage of the intervals $[0, S_{(1)}), (S_{(1)}, S_{(2)}), \dots, (S_{(m-1)}, S_{(m)}), (S_{(m)}, \infty]$ under the function $S(x, \cdot)$ (a function of the second argument with x held fixed), rather than the intervals given explicitly in Lines 12, 20, and 27 of Alg. S1. This algorithm then gives exact coverage for any continuous score function, although it will only be computationally feasible when these preimages can be computed efficiently.

S2. Efficient computation of the full conformal confidence set for ridge regression and Gaussian process regression

A. Ridge regression. When the likelihood of the test input is a function of the prediction from a ridge regression model, it is possible to compute the scores and weights for the full conformal confidence set by fitting $n + 1$ models and $O(n \cdot p \cdot |\mathcal{Y}|)$ additional floating point operations, instead of naively fitting $(n + 1) \times \mathcal{Y}$ models, as demonstrated in Alg. S2.

For the fluorescent protein design experiments, the TestInputLikelihood subroutine in Alg. S2 computed the likelihood in Eq. [6], that is,

$$\begin{aligned}
\textsc{TestInputLikelihood}(a_i + b_i y) &\leftarrow \frac{\exp(\lambda \cdot (a_i + b_i y))}{\cdot \sum_{x \in \mathcal{X}} \exp(\lambda \cdot (C_i + y\mathbf{A}_{-i,n})^T x)}, \\
\textsc{TestInputLikelihood}(a_{n+1}) &\leftarrow \frac{\exp(\lambda \cdot a_{n+1})}{\cdot \sum_{x \in \mathcal{X}} \exp(\lambda \cdot \beta^T x)},
\end{aligned} \tag{S11}$$

where the input space $\mathcal{X}$ was the combinatorially complete set of $8,192$ sequences. The TrainInputLikelihood subroutine returned the likelihood under the training input distribution, which is simply equal to to $1/8192$, since training sequences were sampled uniformly from the combinatorially complete data set. See https://github.com/clarafy/conformal-for-design for an implementation.

Computing the test input likelihoods was dominated by the $(n + 1) \times |\mathcal{Y}|$ normalizing constants, which can be computed efficiently using a single tensor product between an $(n + 1) \times p \times |\mathcal{Y}|$ tensor containing the model parameters, $C_i + y\mathbf{A}_{-i,n}$ and β, and an $|\mathcal{X}| \times p$ data matrix containing all inputs in $\mathcal{X}$. For domains, $\mathcal{X}$, that are too large for the normalizing constants to be computed exactly, one can turn to tractable Monte Carlo approximations.

Algorithm S2 Efficient computation of scores and weights for ridge regression-based feedback covariate shift

Input: training data, $Z_1, \dots, Z_n$, where $Z_i = (X_i, Y_i)$; test input, X_{n+1}; grid of candidate labels, $\mathcal{Y} \subset \mathbb{R}$; subroutine for test input likelihood, TestInputLikelihood$(\cdot)$, that takes an input's predicted fitness and outputs its likelihood under the test input distribution; subroutine for training input likelihood, TrainInputLikelihood$(\cdot)$.
Output: scores $S_i(X_{n+1}, y)$ and likelihood ratios $v(X_i, Z^y_{-i})$ for $i = 1, \dots, n + 1$, $y \in \mathcal{Y}$.

```
for $i = 1, \dots, n$ do
    $C_i \leftarrow \sum_{j=1}^{n-1} Y_{-i;j} \mathbf{A}_{-i;j}$
    $a_i \leftarrow C_i^T X_i$
    $b_i \leftarrow \mathbf{A}_{-i;n}^T X_i$
$\beta \leftarrow (\mathbf{X}^T \mathbf{X} + \gamma I)^{-1} \mathbf{X}^T Y$
$a_{n+1} \leftarrow \beta^T X_{n+1}$
for $i = 1, \dots, n$ do
    for $y \in \mathcal{Y}$ do
        $S_i(X_{n+1}, y) \leftarrow |Y_i - (a_i + b_i y)|$        ▷ Can vectorize via outer product between $(b_1, \dots, b_n)$ and vector of all $y \in \mathcal{Y}$.
        $v(X_i; Z_{-i,y}) \leftarrow$ TestInputLikelihood$(a_i + b_i y)$/TrainInputLikelihood$(X_i)$        ▷ Can vectorize (see commentary on Eq. [S11]).
$S_{n+1}(X_{n+1}, y) \leftarrow |y - a_{n+1}|$
$v(X_{n+1}; Z_{1:n}) \leftarrow$ TestInputLikelihood$(a_{n+1})$/TrainInputLikelihood$(X_{n+1})$
```

B. Gaussian process regression. Here we describe how the scores and weights for the confidence set in Eq. [3] can be computed efficiently, when the likelihood of the test input distribution is a function of the predictive mean and variance of a Gaussian process regression model.

For an arbitrary kernel and two data matrices, $\mathbf{V} \in \mathbb{R}^{n_1 \times p}$ and $\mathbf{V}' \in \mathbb{R}^{n_2 \times p}$, let $K(\mathbf{V}, \mathbf{V}')$ denote the $n_1 \times n_2$ matrix where the (i, j)-th entry is the covariance between the i-th row of $\mathbf{V}$ and j-th row of $\mathbf{V}'$. The mean prediction for X_i of a Gaussian process regression model fit to the i-th augmented LOO data set, $\mu^y_{-i}(X_i)$, is then given by

$$\mu^y_{-i}(X_i) = K(X_i, \mathbf{X}_{-i})[K(\mathbf{X}_{-i}, \mathbf{X}_{-i}) + \sigma^2 I]^{-1} Y^y_{-i},$$

and the model's predictive variance at X_i is

$$K(X_i, X_i) - K(X_i, \mathbf{X}_{-i})[K(\mathbf{X}_{-i}, \mathbf{X}_{-i}) + \sigma^2 I]^{-1} K(\mathbf{X}_{-i}, X_i),$$

where the rows of the matrix $\mathbf{X}_{-i} \in \mathbb{R}^{n \times p}$ are the inputs in Z^y_{-i}, $Y^y_{-i} = (Y_{-i}, y) \in \mathbb{R}^n$ is the vector of labels in Z^y_{-i}, and σ^2 is the (unknown) variance of the label noise, whose value is set as a hyperparameter. Note that the mean prediction is a linear function of the candidate value, y, which is of the same form as the ridge regression prediction in Eq. [5]; furthermore, the predictive variance is constant over y. Therefore, we can mimic Alg. S2 to efficiently compute scores and weights by training just $n + 1$ rather than $(n + 1) \times |\mathcal{Y}|$ models.

S3. Additional details and results on designing fluorescent proteins

Features Each sequence was first represented as a length-thirteen vector of signed bits (-1 or 1), each denoting which of the two wild-type parents the amino acid at a site matches. The features for the sequence consisted of these thirteen signed bits, called the first-order terms in the main text, as well as all $\binom{13}{2}$ products between pairs of these thirteen bits, called the second-order interaction terms.

Additional simulated measurement noise. Each time the i-th sequence in the combinatorially complete data set was sampled, for either training or designed data, we introduced additional simulated measurement noise using the following procedure. Poelwijk et al. (8) found that the Walsh-Hadamard transform of the brightness fitness landscape included up to seventh-order statistically significant terms. Accordingly, we fit a linear model of up to seventh-order terms for each of the combinatorially complete data sets, then estimated the standard deviation of the i-th sequence's measurement noise, σ_i, as the residual between its label and this model's prediction. Each time the i-th sequence was sampled, for either training or designed data, we also sampled zero-mean Gaussian noise with standard deviation σ_i and added it to the i-th sequence's label. This was done to simulate the fact that multiple measurements of the same sequence will yield different labels, due to measurement noise.

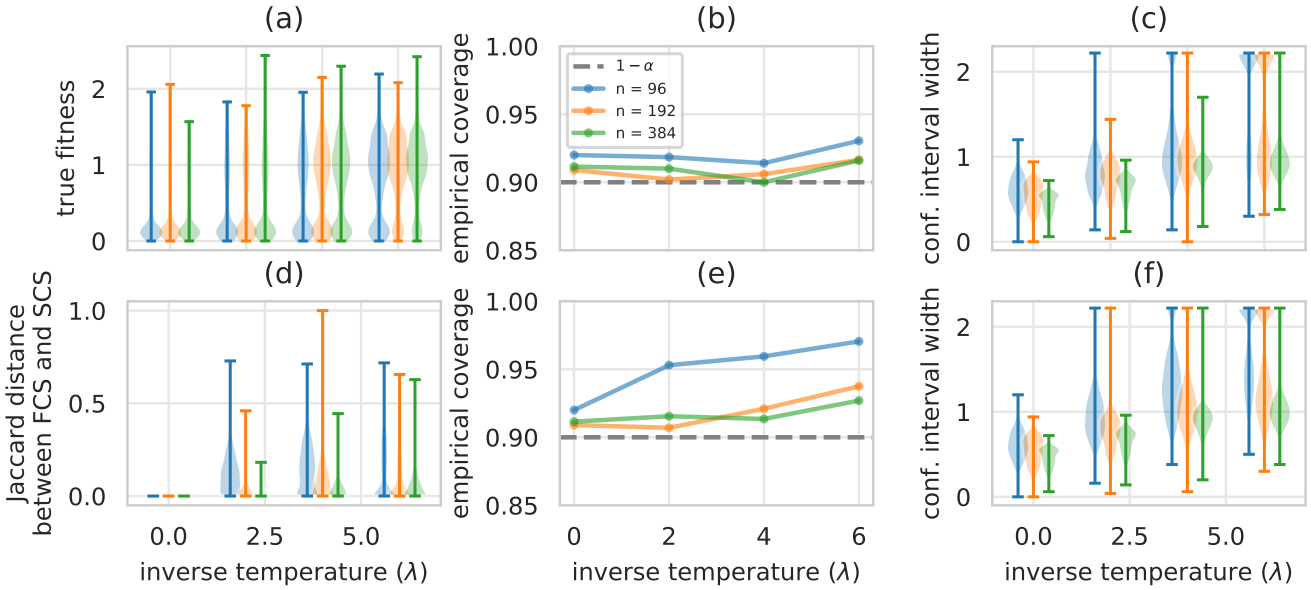


Fig. S2. Quantifying predictive uncertainty for designed proteins, using the red fluorescence data set. (a) Distributions of labels of designed proteins, for different values of the inverse temperature, λ, and different amounts of training data, n. Labels surpass the fitness range observed in the combinatorially complete data set, $[0.025, 1.692]$, due to additional simulated measurement noise. (b) Empirical coverage, compared to the theoretical lower bound of $1 - \alpha = 0.9$ (dashed gray line), and (c) distributions of confidence interval widths achieved by full conformal prediction for feedback covariate shift (our method) over $T = 2000$ trials. (d) Distributions of Jaccard distances between the confidence intervals produced by full conformal prediction for feedback covariate shift and standard covariate shift (1). (e, f) Same as (b, c) but using full conformal prediction for standard covariate shift. In (a), (c), (d), and (f), the whiskers signify the minimum and maximum observed values.

S4. Additional details on AAV experiments

NNK sequence distribution. The NNK sequence distribution is parameterized by independent categorical distributions over the four nucleotides, where the probabilities of the nucleotides are intended to result in a high diversity of amino acids while avoiding stop codons. Specifically, for three contiguous nucleotides corresponding to a codon, the first two nucleotides are sampled uniformly at random from $\{\mathtt{A}, \mathtt{C}, \mathtt{T}, \mathtt{G}\}$, while the last nucleotide is sampled uniformly at random from only $\{\mathtt{T}, \mathtt{G}\}$.

Additional simulated measurement noise. Following Zhu & Brookes et al. (9), the fitness assigned to the i-th sequence was an enrichment score based on its counts before and after a selection experiment, $n_{i,\text{pre}}$ and $n_{i,\text{post}}$, respectively. The variance of this enrichment score for the i-th sequence was estimated as

$$\sigma_i^2 = \frac{1}{n_{i,\text{post}}}\left(1 - \frac{n_{i,\text{post}}}{N_{\text{post}}}\right) + \frac{1}{n_{i,\text{pre}}}\left(1 - \frac{n_{i,\text{pre}}}{N_{\text{pre}}}\right)$$

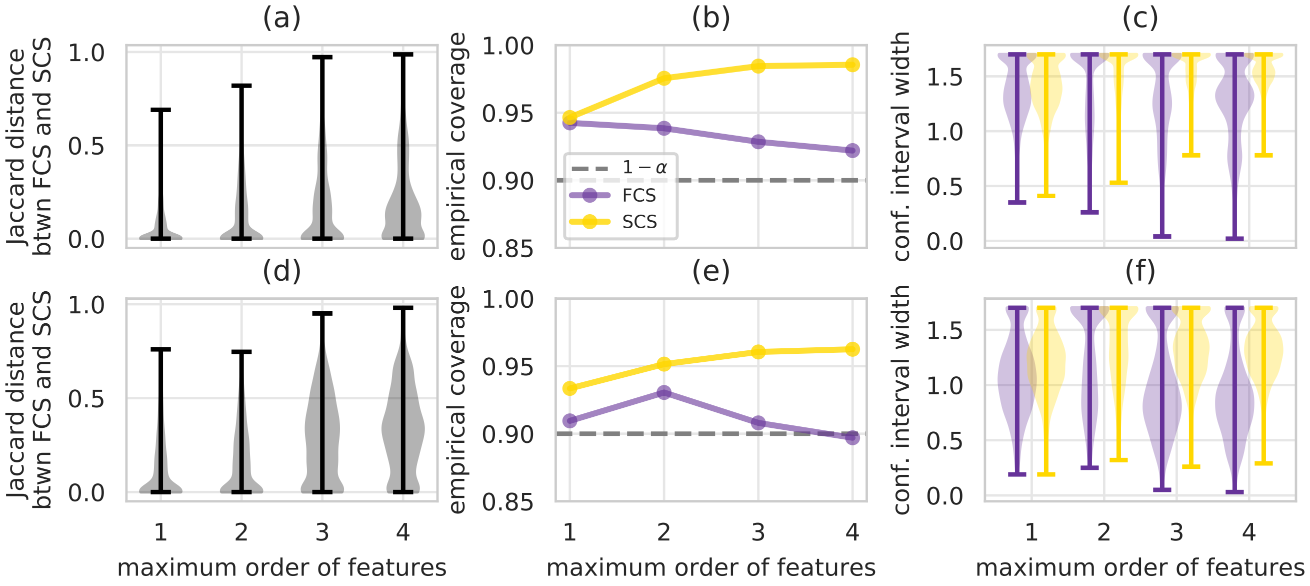


Fig. S3. Quantifying predictive uncertainty for designed proteins using the blue and red fluorescence data sets, for $n = 48$ training data points, $\lambda = 6$, and ridge regression models with features of varying complexity. In particular, the features consist of all interaction terms up to order d between the thirteen sequence sites, where the maximum order, d, is the x-axis of the following subplots. (a) Distributions of Jaccard distances between the confidence intervals produced by conformal prediction for feedback covariate shift (FCS, our method) and standard covariate shift (SCS) (1) for the blue data set over $T = 2000$ trials. (b) Empirical coverage, compared to the theoretical lower bound of $1 - \alpha = 0.9$ (dashed gray line), achieved by conformal prediction for FCS and SCS over those trials. (c) Distributions of confidence interval widths using conformal prediction for FCS and SCS. (d-f) Same as (a-c) but for the red fluorescence data set. In (a), (c), (d), and (f), whiskers signify the minimum and maximum observed values.

where N_{pre} and N_{post} denote the total counts of all the sequences before and after the selection experiment, respectively. Using this estimate, we introduced additional simulated measurement noise to the label of the i-th sequence by adding zero-mean Gaussian noise with a variance of $0.1 \cdot \sigma_i^2$.

Neural network details. As in (9), the neural network took one-hot-encoded sequences as inputs and had an architecture consisting of two fully connected hidden layers, with 100 units each and `tanh` activation functions. It was fit to the $7,552,729$ training data points with the built-in implementation of the Adam algorithm in Tensorflow, using the default hyperparameters and a batch size of 64 for 10 epochs, where each training data point was weighted according to its estimated variance as in (9).

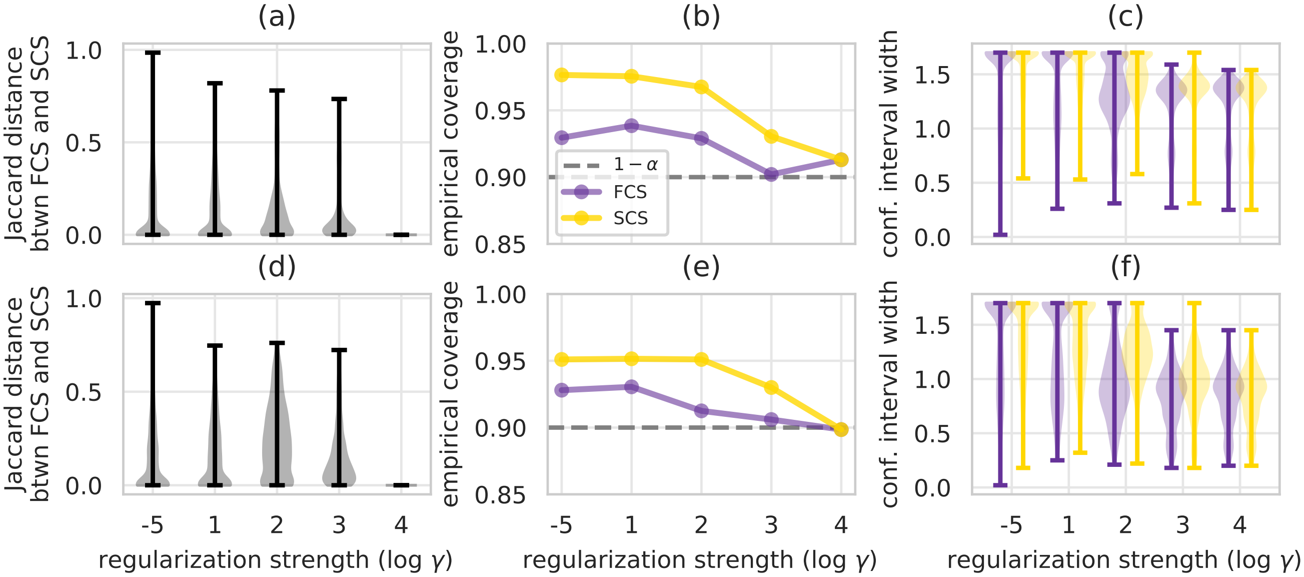


Fig. S4. Quantifying predictive uncertainty for designed proteins using the blue and red fluorescence data sets, for $n = 48$ training data points, $\lambda = 6$, and varying ridge regularization strength, γ. (a) Distributions of Jaccard distances between the confidence intervals produced by conformal prediction for feedback covariate shift (FCS, our method) and standard covariate shift (SCS) (1) for the blue data set over $T = 2000$ trials. (b) Empirical coverage, compared to the theoretical lower bound of $1 - \alpha = 0.9$ (dashed gray line), achieved by conformal prediction for FCS and SCS over those trials. (c) Distributions of confidence interval widths using conformal prediction for FCS and SCS. (d-f) Same as (a-c) but for the red fluorescence data set. In (a), (c), (d), and (f), whiskers signify the minimum and maximum observed values.

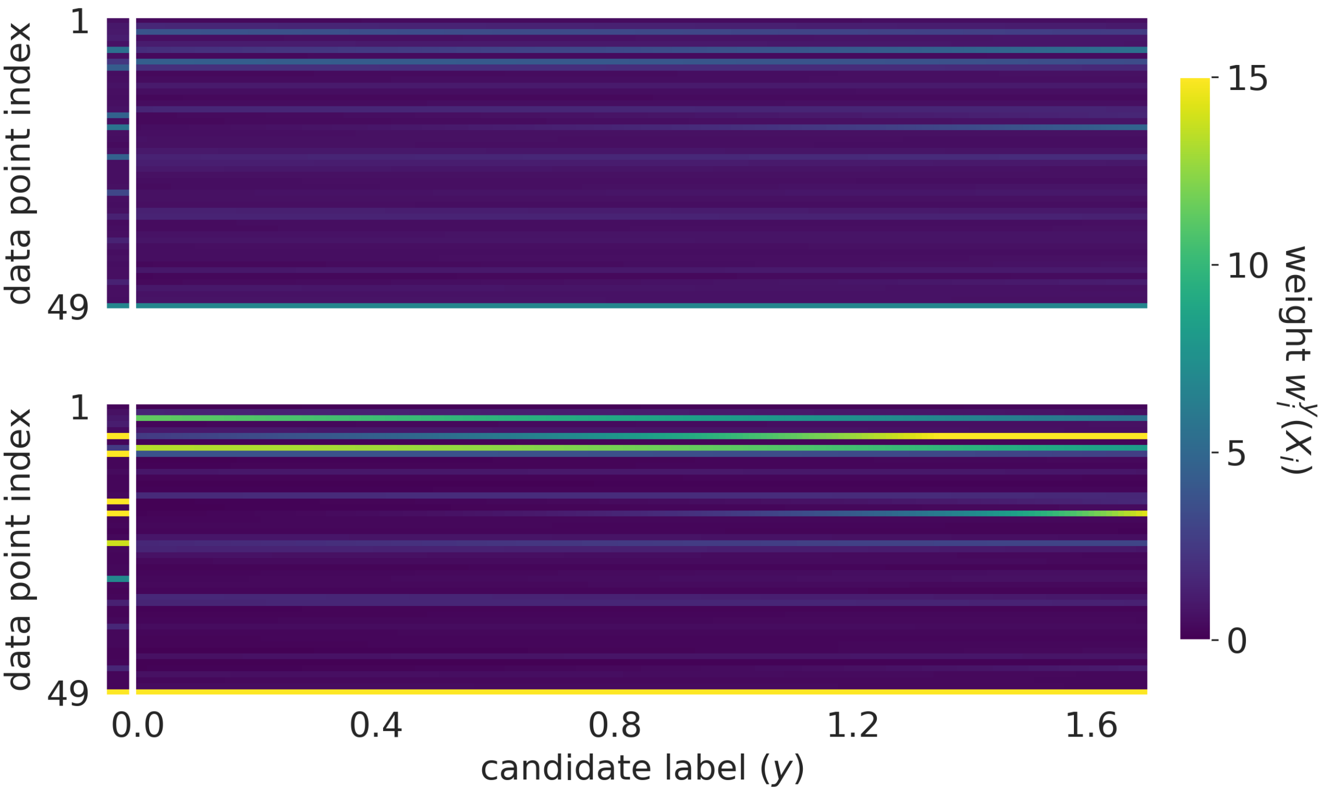


Fig. S5. Comparison between the weights constructed by conformal prediction for feedback covariate shift (FSC, our method) and standard covariate shift (SCS) (1) for one example training data set and resulting designed sequence, for $n = 48$ with the blue fluorescence data set and two different settings of the inverse temperature, λ. Top: For $\lambda = 2$, vector of the $n + 1$ weights prescribed under SCS for the n training data points (data point indices 1 through 48) and the candidate test data points (data point index 49), alongside $(n + 1) \times |\mathcal{Y}|$ matrix of the weights prescribed under FCS for those same $n + 1$ training and candidate test data points. The weight for each of these data points depends on the candidate label, y (x-axis of heatmap), through a linear relationship with y (see Section D). Bottom: same as top but for $\lambda = 6$.

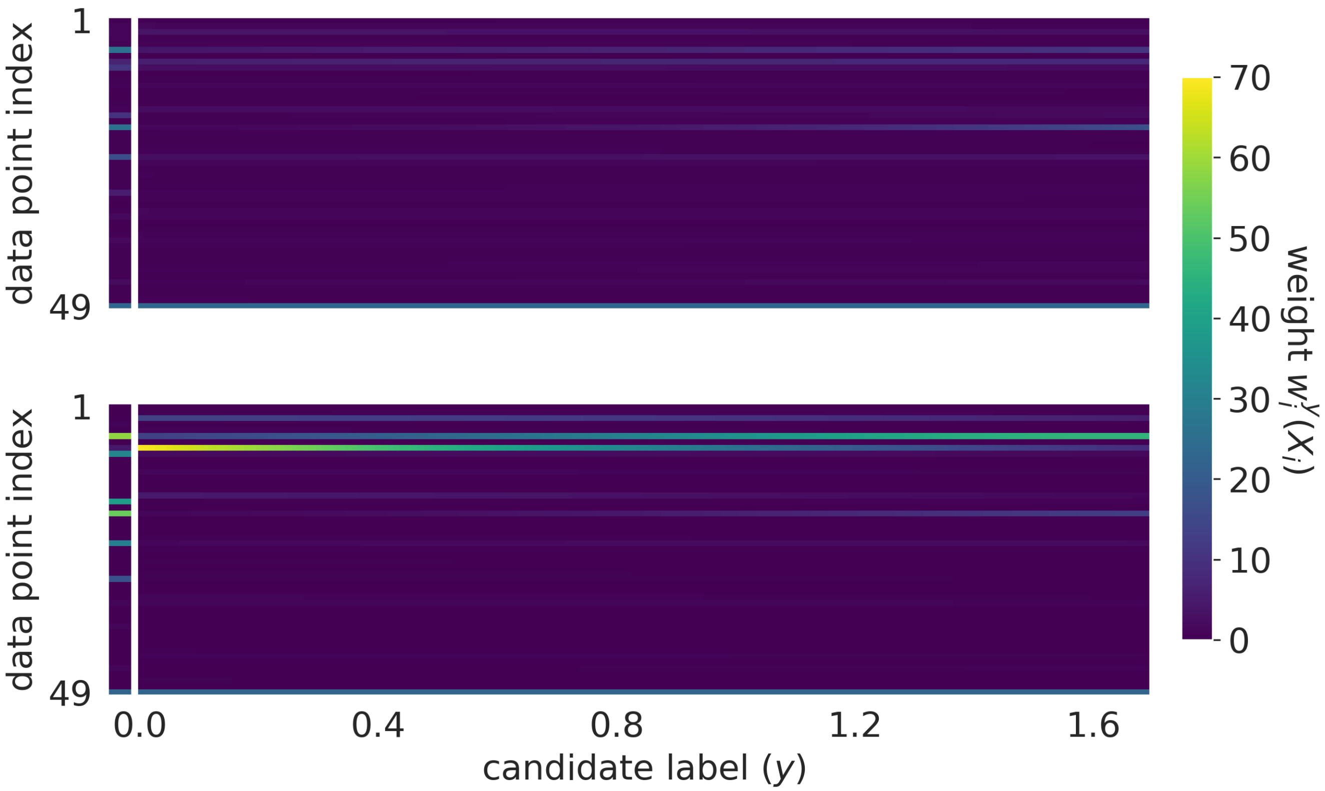


Fig. S6. Comparison between the weights constructed by conformal prediction for feedback covariate shift (FSC, our method) and standard covariate shift (SCS) (1) for one example training data set and resulting designed sequence, for $n = 48$ with the blue fluorescence data set and two different settings of the ridge regularization strength, γ. Top: For $\gamma = 100$, vector of the $n + 1$ weights prescribed under SCS for the n training data points (data point indices 1 through 48) and the candidate test data points (data point index 49), alongside $(n + 1) \times |\mathcal{Y}|$ matrix of the weights prescribed under FCS for those same $n + 1$ training and candidate test data points. The weight for each of these data points depends on the candidate label, y (x-axis of heatmap), through a linear relationship with y (see Section D). Bottom: same as top but for $\gamma = 10$.

References

1. RJ Tibshirani, R Foygel Barber, E Candes, A Ramdas, Conformal prediction under covariate shift in *Advances in Neural Information Processing Systems.* Vol. 32, pp. 2530–2540 (2019).
2. H Papadopoulos, K Proedrou, V Vovk, A Gammerman, Inductive confidence machines for regression in *Machine Learning: European Conference on Machine Learning.* pp. 345–356 (2002).
3. J Lei, M G'Sell, A Rinaldo, RJ Tibshirani, L Wasserman, Distribution-free predictive inference for regression. *Journal of the American Statistical Association* **113**, 1094–1111 (2018).
4. H Shimodaira, Improving predictive inference under covariate shift by weighting the log-likelihood function. *J. Stat. Plan. Inference* **90**, 227–244 (2000).
5. M Sugiyama, KR Müller, Input-dependent estimation of generalization error under covariate shift. *Stat. & Decis.* **23**, 249–279 (2005).
6. M Sugiyama, M Krauledat, KR Müller, Covariate shift adaptation by importance weighted cross validation. *J. Mach. Learn. Res.* **8**, 985–1005 (2007).
7. AN Angelopoulos, S Bates, A gentle introduction to conformal prediction and distribution-free uncertainty quantification. arXiv preprint 2107.07511 (2021).
8. FJ Poelwijk, M Socolich, R Ranganathan, Learning the pattern of epistasis linking genotype and phenotype in a protein. *Nat. Commun.* **10**, 4213 (2019).
9. D Zhu, et al., Optimal trade-off control in machine learning-based library design, with application to adeno-associated virus (aav) for gene therapy. bioRxiv preprint 2021.11.02.467003 (2021).