Transfer Learning Across Fixed-Income Product Classes

Nicolas Camenzind* Damir Filipović†

11 May 2025

Abstract

We propose a framework for transfer learning of discount curves across different fixed-income product classes. Motivated by challenges in estimating discount curves from sparse or noisy data, we extend kernel ridge regression (KR) to a vector-valued setting, formulating a convex optimization problem in a vector-valued reproducing kernel Hilbert space (RKHS). Each component of the solution corresponds to the discount curve implied by a specific product class. We introduce an additional regularization term motivated by economic principles, promoting smoothness of spread curves between product classes, and show that it leads to a valid separable kernel structure. A main theoretical contribution is a decomposition of the vector-valued RKHS norm induced by separable kernels. We further provide a Gaussian process interpretation of vector-valued KR, enabling quantification of estimation uncertainty. Illustrative examples demonstrate that transfer learning significantly improves extrapolation performance and tightens confidence intervals compared to single-curve estimation.

Keywords: yield curve estimation, transfer learning, nonparametric estimator, machine learning in finance, vector-valued reproducing kernel Hilbert space

JEL Classification: C14, E43, G12

1 Introduction

We introduce a framework for transfer learning of discount curves across different fixed-income product classes. Since discount curves are inherently unobservable, they must be inferred from the observable prices of fixed-income instruments. A key feature of the proposed framework is its ability to incorporate complementary market information across product classes. Accurate estimation is critical, as discount curves are fundamental to finance, providing the basis for appropriately discounting future cash flows. Consequently, their precise estimation holds significant practical relevance.

Numerous methods have been proposed for single discount curve estimation. Classical approaches include the parametric Nelson–Siegel–Svensson model [NS87,Sve94,GSW07], as well as nonparametric methods such as Fama–Bliss [FB87], Smith–Wilson [SW01], and Liu–Wu [LW21]. More recently, [FPY24] introduced a kernel ridge regression (KR) framework, providing a theoretically grounded solution based on reproducing kernel Hilbert space (RKHS) theory. KR yields a closed-form, linear estimator and empirically outperforms benchmark models for U.S. and Swiss government bonds [FPY24, CF24]. However, like other methods, KR struggles with extrapolation in maturity ranges where data are sparse or absent [CF24].

*EPFL, Switzerland. Email: nicolas.camenzind@epfl.ch
†EPFL, Switzerland. Email: damir.filipovic@epfl.ch

This limitation motivates the use of transfer learning [WKW16, PY10], a well-established concept in machine learning that is closely related to multitask learning [Car97]. Transfer learning seeks to improve estimation by jointly solving related problems and sharing information across them, particularly when data for the primary task are limited, noisy, or costly to obtain. In the context of discount curves, transfer learning arises naturally across fixed-income product classes, with suitable adjustment for cross-currency effects. A product class refers to a group of fixed-income instruments priced using a common discount curve, denominated in the same currency and characterized by similar risk features such as issuer type, collateralization, or credit quality. Examples include government bonds issued by the same sovereign, interest rate swaps referencing a common overnight risk-free rate [SS19, SIX, MM, ECB, BoEa, Fed], and corporate bonds within a given credit rating class.

This paper develops a theoretical framework for transfer learning of discount curves, complemented by illustrative examples.[1] Our methodology generalizes to any set of fixed-income products that can be represented jointly under a discounted cash flow framework. Although limits to arbitrage may cause different product classes to imply distinct discount curves even when all are considered risk-free [WJ24], we show that the discounted cash flow principle can be naturally embedded into an arbitrage-free pricing framework.

We formulate the transfer learning problem as a vector-valued KR, leading to a convex optimization problem in a vector-valued RKHS. Each component of the solution corresponds to the discount curve implied by a specific product class. Analogous to the scalar case, we derive a closed-form expression for the vector-valued KR estimator.

The theory of vector-valued RKHS is well-established [PR16, MP05], with operator-valued kernels, such as matrix-valued kernels in $\mathbb{R}^n$, playing a central role [KDP+16]. Fundamental results from RKHS theory, including the representer theorem and Moore's theorem, extend naturally to the vector-valued setting [PR16]. A particularly tractable subclass, separable kernels, has been extensively studied [BRBV12, She08, MP04, ARL12]. Separable kernels are constructed as the product of a scalar kernel and a constant covariance matrix, the latter encoding the transfer learning structure. They offer computational advantages, including simple computation of inner products and induced norms [BRBV12].

Building on the scalar case, we introduce an additional regularization term motivated by economic principles, penalizing the spread between discount curves across product classes. A main theoretical contribution of our work is a decomposition of the norm induced by separable kernels, generalizing a result from [BRBV12]. We prove that the resulting regularization yields a valid separable kernel, specifically tailored to our transfer learning problem. Rather than enforcing identical curves, the regularization promotes smoothness of spread curves under an economically motivated norm. This connects naturally to graph regularization techniques [SK03, She08].

We further provide a Gaussian process [CWG19] interpretation of the vector-valued KR, enabling quantification of estimation uncertainty for the discount curves. In doing so, we extend the well-known correspondence between KR and Gaussian processes in the scalar case [RW05] to the setting of transfer learning. Our illustrative examples show that the transfer learning framework significantly tightens confidence intervals around the estimated discount curves.

The remainder of the paper is organized as follows. Section 2 formulates the transfer learning problem for discount curves and presents the representation theorem essential for implementation. Section 3 develops the Gaussian process perspective. Section 4 introduces separable kernels as a natural class of matrix-valued

[1] Two comprehensive empirical studies are in progress, one focusing on government bonds and swaps within the same currency and the other on government bonds across currencies.

kernels for our setting. Section 5 discusses how standard fixed-income products can be embedded into the transfer learning formulation. Section 6 presents illustrative examples, focusing on transfer learning between government bonds and swaps. The appendix provides a self-contained introduction to the theory of vector-valued RKHS, collects all proofs, and details the embedding of the discounted cash flow principle into an arbitrage-free pricing framework. An online appendix includes additional examples.

2 Transfer learning problem formulation

In this section, we present the general problem formulation for transfer learning of discount curves across A different fixed-income product classes. Our framework requires only that the theoretical price of a fixed-income product be expressed as the sum of its discounted cash flows.

Specifically, for every product class $a = 1, \dots, A$, there are M_a fixed-income securities with common cash flow dates $0 < x_1 < \cdots < x_N$, stacked into the column vector $\boldsymbol{x} = (x_1, \dots, x_N)^\top$.[2] The total number of securities is given by $M = M_1 + \cdots + M_A$. For each security we observe noisy ex-coupon prices, $P_a = (P_{a,1}, \dots, P_{a,M_a})^\top$. We denote the associated $M_a \times N$ cash flow matrix by $C_a = (C_{a,ij})$ where $C_{a,ij}$ is the cash flow of security i of product class a that occurs in x_j.

In line with the discounted cash flow principle, we assume that for every product class a, there exists a unique discount curve $g_a : [0, \infty) \to \mathbb{R}$ with $g_a(0) = 1$ and such that the price of every instrument i in product class a is given by

$$P_{a,i} = \sum_{j=1}^{N} C_{a,ij} g_a(x_j). \tag{1}$$

The objective of this paper is to jointly estimate the discount curves $g = (g_1, \dots, g_A)^\top$ from observed market prices P_a. To this end, we decompose each curve g_a as the sum of an exogenous prior function p_a and a hypothesis function h_a, that is,

$$g_a = p_a + h_a \quad \text{for all } a = 1, \dots, A.$$

Here, the prior $p = (p_1, \dots, p_A)^\top : [0, \infty) \to \mathbb{R}^A$ is assumed to satisfy $p(0) = 1$, and the hypothesis $h = (h_1, \dots, h_A)^\top : [0, \infty) \to \mathbb{R}^A$ is constrained to satisfy $h(0) = 0$.[3] A natural and simple choice for the prior is the constant function $p \equiv 1$.

We model h as an element of a vector-valued RKHS $\mathcal{H}$ over the domain $E = [0, \infty)$, taking values in $\mathbb{R}^A$ and satisfying $h(0) = 0$ for all $h \in \mathcal{H}$. The associated reproducing kernel is a matrix-valued function $K : [0, \infty) \times [0, \infty) \to \mathbb{R}^{A \times A}$. Appendix A provides a self-contained introduction to the theory of vector-valued RKHS, including its foundational properties and practical relevance for our setting.

To enable matrix notation, we introduce the following conventions. For any function f, we write $f(\boldsymbol{x}) = (f(x_1), \dots, f(x_N))^\top$ for the corresponding array of function values. For a general matrix $Q \in \mathbb{R}^{m \times n}$, we write $Q_i = (Q_{i1}, \dots, Q_{in})$ for its i-th row vector, and define the vectorization of Q as the vector obtained by stacking its columns, $\operatorname{vec}(Q) = (Q_{11}, \dots, Q_{m1}, Q_{12}, \dots, Q_{m2}, \dots, Q_{1n}, \dots, Q_{mn})^\top \in \mathbb{R}^{nm}$. Accordingly, we

[2] Cash flow dates $\boldsymbol{x}$ are assumed to be common across all product classes without loss of generality.

[3] This additive specification mirrors the structure of linear-rational term structure models; see [FLT17].

denote the matrix $h^\top(\boldsymbol{x}) = (h_1(\boldsymbol{x}), \dots, h_A(\boldsymbol{x})) \in \mathbb{R}^{N\times A}$, and we obtain the vector

$$\operatorname{vec}(h^\top(\boldsymbol{x})) = \begin{pmatrix} h_1(\boldsymbol{x}) \\ \vdots \\ h_A(\boldsymbol{x}) \end{pmatrix} = (h_1(x_1), \dots, h_1(x_N), h_2(x_1), \dots, h_2(x_N), \dots, h_A(x_1), \dots, h_A(x_N))^\top \in \mathbb{R}^{AN}.$$

We also stack the cash flow matrices and price vectors across product classes as

$$\boldsymbol{C} = \begin{pmatrix} C_1 & & \\ & \ddots & \\ & & C_A \end{pmatrix} \in \mathbb{R}^{M\times AN}, \quad \boldsymbol{P} = \begin{pmatrix} P_1 \\ \vdots \\ P_A \end{pmatrix} \in \mathbb{R}^{M},$$

where $\boldsymbol{C}$ is block diagonal with the individual cash flow matrices C_a along the diagonal, and all off-diagonal blocks equal to zero. The discounted cash flow equation (1) then reads $\boldsymbol{P} = \boldsymbol{C}\operatorname{vec}(p^\top(\boldsymbol{x}) + h^\top(\boldsymbol{x}))$. Including pricing errors $\boldsymbol{\epsilon}$ leads to

$$\boldsymbol{P} = \boldsymbol{C}\operatorname{vec}(p^\top(\boldsymbol{x}) + h^\top(\boldsymbol{x})) + \boldsymbol{\epsilon}. \tag{2}$$

Such pricing errors occur due to market imperfections and data errors.

The estimation objective reduces to finding a function $h \in \mathcal{H}$ that balances the tradeoff between the weighted mean-squared pricing error,

$$\sum_{a=1}^{A}\sum_{i=1}^{M_a} \omega_{a,i}\big(P_{a,i} - C_{a,i}p_a(\boldsymbol{x}) - C_{a,i}h_a(\boldsymbol{x})\big)^2,$$

and the regularity of h, as quantified by the vector-valued RKHS norm $\|h\|_{\mathcal{H}}$. The weights $\omega_{a,i} > 0$ are exogenously specified and reflect the relative importance of the pricing terms. This setup corresponds to a vector-valued KR. The regularity penalty depends on the choice of the matrix-valued kernel K, whose structure is encoded in the kernel matrix

$$\boldsymbol{K} = \begin{pmatrix} \boldsymbol{K_{11}} & \cdots & \boldsymbol{K_{1A}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{K_{A1}} & \cdots & \boldsymbol{K_{AA}} \end{pmatrix} \in \mathbb{R}^{AN\times AN}, \tag{3}$$

where each block $\boldsymbol{K_{ab}} \in \mathbb{R}^{N\times N}$ has entries $\boldsymbol{K_{ab},}_{ij} = K_{ab}(x_i, x_j)$. The following theorem formalizes this formulation.

Theorem 2.1. *The unique solution of the vector-valued KR problem*

$$\min_{h\in\mathcal{H}}\Big\{\sum_{a=1}^{A}\sum_{i=1}^{M_a} \omega_{a,i}(P_{a,i} - C_{a,i}p_a(\boldsymbol{x}) - C_{a,i}h_a(\boldsymbol{x}))^2 + \lambda\|h\|_{\mathcal{H}}^2\Big\} \tag{4}$$

is given by $\bar{h} = \sum_{j=1}^{N} K(\cdot, x_j)\beta_j$ *where* $\beta = (\beta_1, \dots, \beta_N) \in \mathbb{R}^{A\times N}$ *takes the form*

$$\operatorname{vec}(\beta^\top) = \boldsymbol{C}^\top\left(\boldsymbol{C}\boldsymbol{K}\boldsymbol{C}^\top + \boldsymbol{\Lambda}\right)^{-1}(\boldsymbol{P} - \boldsymbol{C}\operatorname{vec}(p^\top(\boldsymbol{x}))),$$

for the block diagonal matrix $\boldsymbol{\Lambda} = \operatorname{diag}(\Lambda_1, \dots, \Lambda_A) \in \mathbb{R}^{M\times M}$ *with* $\Lambda_a = \operatorname{diag}(\lambda/\omega_{a,1}, \dots, \lambda/\omega_{a,M_a})$. *The*

corresponding discount curves are given by $\bar{g} = p + \bar{h}$.

A common choice for the weights $\omega_{a,i}$ is based on the duration of the underlying securities; see Section 6.2 for details. The choice of the kernel K is critical, as it should be both economically meaningful and computationally tractable. In our implementation, we use a separable kernel of the form $K(x, y) = B\,k(x, y)$, where $B \in \mathbb{R}^{A\times A}$ is a symmetric positive semi-definite matrix and $k : [0, \infty) \times [0, \infty) \to \mathbb{R}$ is a scalar-valued kernel. This form decouples the dependence on product class, captured by the matrix B, from the time-to-maturity dependence, captured by k. For further background on separable kernels and their properties, we refer to Appendix A.

Remark 2.2. *Theorem 2.1 can be extended towards infinite weights* $\omega_{a,i} = \infty$*, with the convention* $\lambda/\infty = 0$*, which corresponds to an exact fit of* $P_{a,i}$*, for selected* a, i*. This requires that the corresponding block of* $\boldsymbol{C}\boldsymbol{K}\boldsymbol{C}^\top$ *is invertible. See [FPY24, Theorem A.1] for details.*

Remark 2.3. *Theorem 2.1 remains valid even when no quotes are available for a given product class* a*, i.e., when* $M_a = 0$*. In this case, the corresponding rows in* $\boldsymbol{C}$ *and* $\boldsymbol{P}$ *are omitted, and we adopt the convention that* $\sum_{i=1}^{0} = 0$*. Remarkably, the solution curve* $\bar{h}_a$ *still depends on the other product classes via the joint regularization term. In the extreme case where no quotes are available at all,* $M = 0$*, the solution* $\bar{h}$ *is identically zero, and the resulting discount curve reduces to the prior,* $\bar{g} = p$*.*

3 Gaussian process view

Similar to the scalar case one can develop a Gaussian process perspective of the kernel ridge regression in the vector-valued case. We first discuss the general case and then specialize to separable kernels.

3.1 Vector-valued Gaussian processes

We recap the theory of vector-valued Gaussian processes and prove the equivalence of the posterior mean function and the vRK solution. We denote by $\mathcal{N}(m, \Sigma)$ the multivariate normal distribution with mean vector m and covariance matrix Σ.

Definition 3.1 (vector-valued Gaussian process)**.** *We say* $g : E \to \mathbb{R}^A$ *is a vector-valued Gaussian process with mean function* $m = (m_1, \dots, m_A)^\top : E \to \mathbb{R}^A$ *and kernel function* $K(x, y) : E \times E \to \mathbb{R}^{A\times A}$ *if and only if for any* $\boldsymbol{x} = (x_1, \dots, x_N)^\top$

$$\operatorname{vec}(g^\top(\boldsymbol{x})) \sim \mathcal{N}(\operatorname{vec}(m^\top(\boldsymbol{x})), \boldsymbol{K})$$

with $m^\top(\boldsymbol{x}) = (m_1(\boldsymbol{x}), \dots, m_A(\boldsymbol{x})) \in \mathbb{R}^{N\times A}$ *and* $\boldsymbol{K}$ *as in (3). In this case we write* $g \sim \mathcal{MG}(m, K)$*.*

Remark 3.2. *There is no restriction to use* $\boldsymbol{x}$ *across all components of* g*. One can formulate a Gaussian process for any finite collection of points* $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$*,* $\boldsymbol{x}_i \in \mathbb{R}^N$*, such that* $(g_1(\boldsymbol{x}_1), \dots, g_A(\boldsymbol{x}_n)) \in \mathbb{R}^{N\times A}$*.*

We replicate the results [FPY24, Section A.4] for the vector-valued case which is straightforward. For this we assume that g is a vector-valued Gaussian process with mean function m and kernel function $K(x, y)$, i.e., $g \sim \mathcal{MG}(m, K)$. The pricing equation with errors is given by equation (2) where we assume $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$

with

$$\boldsymbol{\Sigma} = \begin{pmatrix} \Sigma_1 & 0 & \dots & 0 \\ 0 & \Sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \Sigma_A \end{pmatrix} \in \mathbb{R}^{M \times M}$$

for symmetric positive definite $M_a \times M_a$-matrices Σ_a.

For n arbitrary cash flow dates $\boldsymbol{z} = (z_1, \dots, z_n)^\top$ this implies that $\operatorname{vec}(g^\top(\boldsymbol{z}))$ and $\boldsymbol{P}$ are jointly Gaussian distributed

$$\begin{pmatrix} \operatorname{vec}(g^\top(\boldsymbol{z})) \\ \boldsymbol{P} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \operatorname{vec}(m^\top(\boldsymbol{z})) \\ \boldsymbol{C} \operatorname{vec}(m^\top(\boldsymbol{x})) \end{pmatrix}, \begin{pmatrix} K(\boldsymbol{z}, \boldsymbol{z}^\top) & K(\boldsymbol{z}, \boldsymbol{x}^\top)\boldsymbol{C}^\top \\ \boldsymbol{C}K(\boldsymbol{x}, \boldsymbol{z}^\top) & \boldsymbol{C}\boldsymbol{K}\boldsymbol{C}^\top + \boldsymbol{\Sigma} \end{pmatrix} \right) \tag{5}$$

where $K(\boldsymbol{x}, \boldsymbol{z}^\top)$ is the block matrix with entries $K(x_i, z_j)$, similar for $K(\boldsymbol{z}, \boldsymbol{z}^\top)$ such that $\boldsymbol{K} = K(\boldsymbol{x}, \boldsymbol{x}^\top)$.

Bayesian updating implies that the conditional distribution of g, given the observed prices $\boldsymbol{P}$, is still vector-valued Gaussian with posterior mean function

$$m^{\text{post}}(z) = m(z) + K(z, \boldsymbol{x}^\top) \operatorname{vec}(\beta^\top), \tag{6}$$

with

$$\operatorname{vec}(\beta^\top) = \boldsymbol{C}^\top(\boldsymbol{C}\boldsymbol{K}\boldsymbol{C}^\top + \boldsymbol{\Sigma})^{-1}(\boldsymbol{P} - \boldsymbol{C} \operatorname{vec}(m^\top(\boldsymbol{x})), \tag{7}$$

and posterior kernel function

$$K^{\text{post}}(y, z) = K(y, z) - K(y, \boldsymbol{x}^\top)\boldsymbol{C}^\top(\boldsymbol{C}\boldsymbol{K}\boldsymbol{C}^\top + \boldsymbol{\Sigma})^{-1}\boldsymbol{C}K(\boldsymbol{x}, z).$$

Hence we recovered the following vector-valued version of [FPY24, Lemma 9].

Theorem 3.3. *Suppose the kernel K, the prior mean function $m = p$ and $\boldsymbol{\Sigma} = \boldsymbol{\Lambda}$ are as in Theorem 2.1. Then the posterior mean function (6) coincides with the KR estimator $\bar{g}(z)$ in Theorem 2.1.*

The posterior mean is invariant with respect to scaling of K and $\boldsymbol{\Sigma}$ by a factor $s > 0$. That is to replace K by $K' = sK$ and $\boldsymbol{\Sigma}' = s\boldsymbol{\Sigma}$. Similar as in [FPY24] one can use (5) to derive at a prior log-likelihood function of s given prices $\boldsymbol{P}$

$$\mathcal{L}(s) = -q_2 \frac{1}{s} - \frac{M}{2} \log(s) - q_1$$

for $q_2 = \frac{1}{2}(\boldsymbol{P} - \boldsymbol{C} \operatorname{vec}(m^\top(\boldsymbol{x}))^\top(\boldsymbol{C}\boldsymbol{K}\boldsymbol{C}^\top + \boldsymbol{\Sigma})^{-1}(\boldsymbol{P} - \boldsymbol{C} \operatorname{vec}(m^\top(\boldsymbol{x})))$ and $q_1 = \frac{1}{2}\log|\boldsymbol{C}\boldsymbol{K}\boldsymbol{C}^\top + \boldsymbol{\Sigma}| + \frac{M}{2}\log(2\pi)$. The maximum log-likelihood is attained for

$$\hat{s} = \frac{2q_2}{M}.$$

Remark 3.4. *When $\boldsymbol{K_{ab}} = 0$ for all $a \neq b$ the posteriori mean estimator corresponds to $A > 1$ independent scalar learned mean estimators. This can be seen from (7) as in this case the block diagonal structure of $\boldsymbol{K}$ factors through, given that the matrices $\boldsymbol{C}$ and $\boldsymbol{\Sigma}$ are blockdiagonal by definition. However, the confidence bands might differ as $\hat{s}$ does. In the scalar case the optimal scaling is given by $\hat{s}_a = \frac{2q_{2,a}}{M_a}$, for the respective value $q_{2,a}$. On the other hand, for the transfer learning case it holds by definition $M = \sum_a M_a$, and $\boldsymbol{K_{ab}} = 0$ implies $q_2 = \sum_a q_{2,a}$. Hence in general the scaling factors differ, $\frac{q_{2,a}}{M_a} \neq \frac{\sum_b q_{2,b}}{\sum_b M_b}$, for individual classes a.*

3.2 Gaussian process view for separable kernels

The Gaussian process view reveals some additional interpretation for separable kernels. In particular, one can use the theory of Gaussian matrix variate distributions to get some additional insights how different components of g are correlated to each other. The key findings are given below. We first recall the definition and some basic properties of matrix variate Gaussian distributions.

Definition 3.5. *The random matrix $X \in \mathbb{R}^{N\times A}$ is said to have a matrix variate Gaussian distribution with mean matrix $M \in \mathbb{R}^{N\times A}$, covariance matrices $\Sigma \in \mathbb{R}^{N\times N}$ and $B \in \mathbb{R}^{A\times A}$ if and only if the probability density function is given by*

$$p(X|M,\Sigma,B) = (2\pi)^{-\frac{AN}{2}} (\det\Sigma)^{-\frac{A}{2}} (\det B)^{-\frac{N}{2}} \exp\left(-\frac{1}{2}\operatorname{tr}\left(B^{-1}(X-M)^\top \Sigma^{-1}(X-M)\right)\right)$$

We denote a matrix variate Gaussian distributed X as $X \sim \mathcal{MN}(M,\Sigma,B)$.

It holds that $X \sim \mathcal{MN}(M,\Sigma,B)$ if and only if $\operatorname{vec}(X) \sim \mathcal{N}(\operatorname{vec}(M), B\otimes\Sigma)$, see [CWG19, Theorem 2]. This again implies that the transpose $X^\top \sim \mathcal{MN}(M^\top, B, \Sigma)$, see [CWG19, Theorem 1]. Hence, in view of Definition 3.1, for a separable kernel $K(x,y) = Bk(x,y)$, we have that $g \sim \mathcal{MG}(m,K)$ is equivalent to $g^\top(\boldsymbol{x}) \sim \mathcal{MN}(m^\top(\boldsymbol{x}), \boldsymbol{k}, B)$ for $\boldsymbol{K} = B\otimes\boldsymbol{k}$ and $m^\top(\boldsymbol{x}) = (m_1(\boldsymbol{x}),\dots,m_A(\boldsymbol{x}))$, and where $\boldsymbol{k}$ denotes the matrix with entries $\boldsymbol{k}_{ij} = k(x_i,x_j)$.

This leads to a natural interpretation of the variance and covariance structure of the discount curves. From the above, we obtain $\operatorname{Var}(g_a(x)) = B_{aa}k(x,x)$ and $\operatorname{Cov}(g_a(x),g_b(y)) = B_{ab}k(x,y)$. The separable kernel structure allows us to interpret each entry B_{ab} as the covariance between product classes a and b, scaled by the scalar kernel $k(x,y)$, which reflects the time-to-maturity effect and is independent of the product class. The correlation is obtained by normalization. More details and a general decomposition result for matrix-valued kernels are provided in Lemmas A.8 and A.9 in the appendix.

4 A workable class of separable kernels

In this section, we introduce the baseline model used for both theoretical analysis and empirical implementation. The formulation is motivated by economic reasoning and results in a tractable optimization problem with a closed-form solution. It extends the single-curve model of [FPY24] to the vector-valued case. We proceed in two steps. First, we heuristically construct a joint estimation objective that includes spread penalties between curves. Second, we show that the resulting problem is equivalent to a vector-valued KR with a separable matrix-valued kernel.

We begin with A scalar-valued estimation problems, each for a fixed-income product class $a = 1,\dots,A$,

$$\min_{h_a\in\mathcal{H}_k} \sum_{i=1}^{M_a} \omega_{a,i}\big(P_{a,i} - C_{a,i}p_a(\boldsymbol{x}) - C_{a,i}h_a(\boldsymbol{x})\big)^2 + \gamma_a\|h_a\|^2_{\mathcal{H}_k},$$

where k is a common scalar kernel with RKHS $\mathcal{H}_k$, and $\gamma_a > 0$ is the regularity parameter for class a. Each problem yields an individual estimator h_a. Estimating the A curves independently is equivalent to solving the joint optimization problem

$$\min_{h_1,\dots,h_A\in\mathcal{H}_k} \sum_{a=1}^{A}\left\{\sum_{i=1}^{M_a}\omega_{a,i}(P_{a,i} - C_{a,i}p_a(\boldsymbol{x}) - C_{a,i}h_a(\boldsymbol{x}))^2 + \gamma_a\|h_a\|^2_{\mathcal{H}_k}\right\}.$$

This can be viewed as a single objective over the product space $(\mathcal{H}_k)^A$.

To introduce dependencies across product classes, we extend the regularization to the differences between curves. Specifically, we add spread penalties of the form

$$\sum_{a=1}^{A}\sum_{b>a}\Theta_{ab}\|h_a - h_b\|^2_{\mathcal{H}_k},$$

where $\Theta_{ab} \geq 0$ controls the strength of transfer learning between classes a and b.[4] These terms encourage similarity between curves without forcing equality. Instead, they penalize irregularities in the spread curves through the RKHS norm $\|\cdot\|_{\mathcal{H}_k}$. We use the terms regularity and smoothness interchangeably, referring specifically to the notion of smoothness induced by the RKHS norm $\|\cdot\|_{\mathcal{H}_k}$. In [FPY24], a specific kernel k is proposed that encodes economically meaningful smoothness properties for discount curves. We adopt this kernel specification in our implementations, see (14) and (15) below.

The complete transfer learning problem is

$$\min_{h_1,\ldots,h_A\in\mathcal{H}_k}\sum_{a=1}^{A}\Big\{\sum_{i=1}^{M_a}\omega_{a,i}(P_{a,i} - C_{a,i}p_a(\boldsymbol{x}) - C_{a,i}h_a(\boldsymbol{x}))^2 + \gamma_a\|h_a\|^2_{\mathcal{H}_k} + \sum_{b>a}\Theta_{ab}\|h_a - h_b\|^2_{\mathcal{H}_k}\Big\}. \tag{8}$$

The following theorem shows that (8) can be interpreted as a vector-valued KR with a separable kernel. This implies in particular that the optimization problem is convex and admits a unique solution.

Theorem 4.1. *Let $\Theta_{ab} \geq 0$ for $a < b$, and define $\Theta_{ba} = \Theta_{ab}$. Then the transfer learning problem (8) is equivalent to the vector-valued KR problem (4) with $\lambda = 1$ and vector-valued RKHS norm*

$$\|h\|^2_{\mathcal{H}} = \sum_{a=1}^{A}\gamma_a\|h_a\|^2_{\mathcal{H}_k} + \sum_{a=1}^{A}\sum_{b>a}\Theta_{ab}\|h_a - h_b\|^2_{\mathcal{H}_k} \tag{9}$$

which corresponds to the separable reproducing kernel $K(x,y) = Bk(x,y)$, where $B = Q^{-1}$ and $Q \in \mathbb{R}^{A\times A}$ is defined by

$$Q_{ab} = \begin{cases} \gamma_a + \sum_{j\neq a}\Theta_{aj}, & \text{if } a = b, \\ -\Theta_{ab}, & \text{if } a \neq b. \end{cases} \tag{10}$$

The parameters Θ_{ab} may be interpreted as edge weights on a graph with A nodes, each corresponding to a product class. The matrix Q equals the sum of the diagonal matrix of γ_a and the Laplacian of the graph, $Q = \mathrm{diag}(\gamma) + L(\Theta)$. This formulation is known as graph regularization in the literature, see [BRBV12] and [She08].

5 Standard fixed-income products

This section shows how standard fixed-income instruments can be expressed in the discounted cash flow format (1), thereby enabling application of our estimation framework. While this formulation may lead

[4] We also considered adjusting the individual regularization parameters γ_a downward to keep the total regularization weight constant when adding spread penalties. This corresponds to choosing $\lambda < 1$ in (4). However, in our empirical studies we found that such scaling can introduce irregularities in the estimated discount curves, which is undesirable. We therefore recommend keeping the values of γ_a fixed and setting $\lambda = 1$ to achieve a well-balanced and effective transfer learning outcome, as stated in Theorem 4.1.

to distinct discount curves across different product classes, we demonstrate in Appendix C how, under an arbitrage-free pricing framework, a single risk-free curve and the corresponding function g may be recovered.

We proceed as follows: we first show how coupon bonds can be cast into the pricing format (1), then extend this formulation to fixed–floating interest rate swaps. Finally, we examine cross-currency swaps and show how transfer learning facilitates joint estimation of discount curves and forward exchange rates, offering insights into multi-currency pricing.

5.1 Coupon bonds

The transformation of fixed-coupon bonds into the discounted cash flow format (1) is straightforward. Consider a bond with notional normalized to one and coupons $c_1, \dots, c_n$ paid at dates $0 < T_1 < \cdots < T_n$, where T_n denotes the bond's maturity at which the notional is paid.[5] Assuming the bond is default-free, the price is given by

$$P = \sum_{j=1}^{n} c_j g(T_j) + g(T_n).$$

Defaultable bonds are treated in Appendix C under an arbitrage-free pricing framework.

5.2 Interest rate swaps

We consider a standard fixed–floating interest rate swap with start (first reset) date $T_0 \geq 0$ and maturity date T_n. We denote the reset and cash flow dates of the fixed payments leg by $T_0 < T_1 < \cdots < T_n$ and of the floating payments leg by $T_0 = t_0 < t_1 < \cdots < t_m = T_n$. For simplicity, the accrual periods along both legs are assumed to be constant and denoted by $\Delta = T_i - T_{i-1}$ and $\delta = t_i - t_{i-1}$, respectively.[6] The swap is spot starting when $T_0 = 0$ and forward starting when $T_0 > 0$.

The present values of the fixed and floating legs are given by

$$PV_{\text{fixed}} = \Delta R \sum_{i=1}^{n} g(T_i),$$
$$PV_{\text{floating}} = g(T_0) - g(T_n),$$

where R denotes the corresponding fixed swap rate. At inception, the swap has zero value, so that $PV_{\text{floating}} = PV_{\text{fixed}}$. We bring this into the desired format (1) as follows. For a spot-starting swap, $T_0 = 0$, the price is set to $P = 1$, which gives

$$1 = g(T_n) + \Delta R \sum_{i=1}^{n} g(T_i). \tag{11}$$

For a forward-starting swap, $T_0 > 0$, the price is set to $P = 0$, which gives

$$0 = g(T_n) - g(T_0) + \Delta R \sum_{i=1}^{n} g(T_i). \tag{12}$$

[5] The generic time grid (x_i) used in (1) is assumed to be fine enough to cover all potential cash flow dates across product classes. Hence, most entries in each row of $\boldsymbol{C}$ are zero.

[6] This can be generalized to specific day count conventions for both legs where the accrual periods depend on the actual dates, replacing the constant Δ and δ by $\Delta(T_{i-1}, T_i)$ and $\delta(t_{j-1}, t_j)$, respectively.

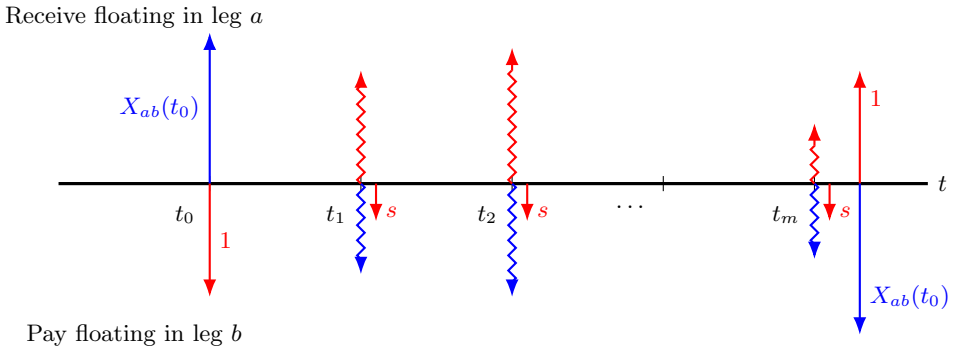
5.3 Cross-currency swaps

Cross-currency swaps (XCCY) involve cash flows in two currencies and combine features of both interest rate and foreign exchange (FX) instruments. See, e.g., [Ran23, BHJ+19] for introductions. We denote the spot exchange rate prevailing at time t as $X_{ab}(t)$, defined as the price of one unit of (base) currency a in terms of (quote) currency b.

A typical use case involves a domestic, say Swiss, firm that holds CHF and wishes to buy a USD-denominated bond. To hedge currency risk, the firm enters a XCCY swapping USD coupon payments against CHF cash flows.[7]

The most standardized and actively traded XCCY is the floating–floating type used in the interbank market. At the start date t_0, notional amounts in both currencies are exchanged at the prevailing spot exchange rate $X_{ab}(t_0)$. Thereafter, floating interest payments are made in each currency at dates $t_0 < t_1 < \cdots < t_m$, typically quarterly and based on overnight risk-free rates (RFRs). The initial notional amounts are re-exchanged at maturity date t_m. A basis spread s is typically added to the less liquid currency leg to reflect liquidity differences and funding imbalances between the two currencies. Figure 1 illustrates this.

Figure 1: Schematic cash flows of a floating–floating XCCY swap



The figure shows the cash flow diagram of a floating–floating XCCY swap. We take the view of receiving leg a while making periodically payments in leg b. Thus, downward pointed arrows reflect a cash flow we have to pay. The wiggled lines denote the floating payments. Straight line are the exchange of notionals and basis spread payments. We assume the basis spread s is added on leg a. It is common that this spread is negative, indicated by the downward pointed arrow. We use the convention to normalize the notional of leg a to 1 so that the corresponding notional of leg b is given by $X_{ab}(t_0)$.

An additional feature common in interbank markets is mark-to-market (MTM) resets of the notional leg. These reduce counterparty risk but, as shown in Appendix C, have no impact on present values.

End clients generally prefer fixed interest payments. Banks accommodate this by combining floating—floating XCCY with standard fixed–floating interest rate swaps. This composite structure is also necessary for estimating the discount curve using our framework. We focus on the non-liquid leg (currency a), and bring this now into the desired format (1). Thereto, let R_a be the fixed swap rate of a standard RFR-based swap in currency a with the same maturity as the XCCY. We assume this rate is observable from the market.[8] More specifically, let $t_0 = T_0 < T_1 < \cdots < T_n = t_m$ be the fixed leg's payment dates with constant accrual period $\Delta = T_i - T_{i-1}$, and assume that currency b is the more liquid leg, so the basis spread s is added to leg a. For a spot-starting XCCY, $T_0 = 0$, we set $P = 1$. The corresponding discounted cash flow

[7] Another example is two firms located in different countries with different currencies. Each exhibits cheaper local funding sources. To raise funds abroad they can enter into a bilateral XCCY.

[8] This is standard practice in well-developed swap markets.

equation becomes

$$1 = g_{a:b}(T_n) + \Delta R_a \sum_{i=1}^{n} g_{a:b}(T_i) + \delta s \sum_{j=1}^{m} g_{a:b}(t_j).$$

For a forward-starting XCCY, $T_0 > 0$, the price is set $P = 0$ and we obtain

$$0 = g_{a:b}(T_n) - g_{a:b}(T_0) + \Delta R_a \sum_{i=1}^{n} g_{a:b}(T_i) + \delta s \sum_{j=1}^{m} g_{a:b}(t_j).$$

Here, $g_{a:b}(\cdot)$ denotes the discount curve for currency a induced by currency b via an XCCY. It incorporates the cross-currency basis and is generally distinct from the discount curve $g_a(\cdot)$ that corresponds to standard interest rate swaps in currency a. If the basis spread s is zero, $g_{a:b}(\cdot)$ coincides with $g_a(\cdot)$.

An important byproduct of this formulation is an expression for the forward exchange rate that incorporates the cross-currency basis. Let $F_{ab}(x)$ denote the forward exchange rate fixed at time 0 for maturity x. Then

$$F_{ab}(x) = X_{ab}(0)\frac{g_{a:b}(x)}{g_b(x)}. \tag{13}$$

This identity can be derived by considering a spot-starting XCCY in combination with an interest rate swap with a single payment at $t_1 = T_1 = x$. Investing one unit of currency a via this XCCY and swapping the floating payment for fixed results in a fixed payoff of $\frac{1}{g_{a:b}(x)}$ units of currency a at maturity x. Alternatively, the same initial amount can be used to purchase $\frac{X_{ab}(0)}{g_b(x)}$ units of the discount bond with maturity x in currency b. At maturity, this yields a cash flow in currency b, which is then converted back into currency a at the forward exchange rate $F_{ab}(x)$, resulting in a payoff of $\frac{X_{ab}(0)}{g_b(x)F_{ab}(x)}$ in currency a. Since both strategies yield deterministic payoffs, the absence of arbitrage implies that they must be equal, which proves (13).

Remark 5.1. *Textbook covered interest parity (CIP) posits that $F_{ab}(x) = X_{ab}(0)\frac{g_a(x)}{g_b(x)}$. However, in practice, deviations from CIP are persistent, as $g_{a:b}(x) \neq g_a(x)$ due to liquidity differences and funding constraints. Most currencies exhibit a negative basis against USD, meaning counterparties are willing to accept a lower return to obtain USD funding, which manifests as $s < 0$ in observed XCCY swaps.*

6 Implementation and examples

This section illustrates the effects of transfer learning through a series of representative examples. While not intended as a full empirical study, these examples demonstrate the model's behavior in realistic scenarios. We focus on transfer learning across $A = 2$ product classes, government bond and RFR-based swaps within the same currency, for four currencies. We first introduce the data, followed by a description of the base model and implementation details. We conclude with an illustration of the effects of transfer learning.

6.1 Data

We source data of government bonds and RFR-based swaps from Bloomberg[9] for four currencies, CHF, EUR, GBP, and USD, on five representative business days of the years 2020 to 2024. Following the approach in [CF24], we select the mid-June (nearest available) business day of each year: 2020-06-15, 2021-06-15, 2022-06-15, 2023-06-15, and 2024-06-14.

[9] Bloomberg Finance L.P.

For government bonds, we use dirty prices, assuming same-day settlement. Our selection includes all fully taxable, non-callable coupon bonds, applying filters consistent with [FB87, LW21]. Following [GSW07], we exclude bonds with fewer than 90 days to maturity and omit all bills. The CHF market has the smallest number of bonds, while the USD market has the largest. Maturity distributions vary across markets. CHF and GBP markets contain bonds with maturities over 40 and 50 years, respectively. EUR and USD maturities initially extend to about 35 years, tapering to around 30 years. Overall, maturity coverage is denser in USD and EUR.

Swap data consists of rates for standard fixed–floating interest rate swaps, where the floating leg is linked to transaction-based overnight RFRs [SS19]. For Switzerland, the reference rate is the Swiss Average Rate Overnight (SARON) [SIX], which is collateralized. In the euro area, two benchmarks exist: EURIBOR and the Euro Short-Term Rate (ESTR) [ECB]. We use ESTR, which is unsecured but still considered risk-free. In the UK, the reformed RFR is the Sterling Overnight Index Average (SONIA) [BoEb], also unsecured. In the U.S., the Secured Overnight Financing Rate (SOFR) [Fed] serves as the RFR and is collateralized. The number of available swap tenors is broadly comparable across currencies, ranging from overnight to 50 years. Both ESTR and SONIA tenors even extend to 60 years.

6.2 Base model and implementation details

To implement the KR estimator in (4), we adopt duration-based weights ω_i, as commonly used in the literature; see, e.g., [FPY24, CF24]. For any fixed-income instrument i with cash flows C_{ij} at dates x_j, its price as a function of yield-to-maturity (YTM) Y is given by

$$Y \mapsto \Pi_i(Y) = \sum_{j=1}^{n} C_{ij} e^{-Y x_j}.$$

The market-implied YTM Y_i is defined by $\Pi_i(Y_i) = P_i$, where P_i denotes the observed market price. The model-implied YTM Y_i^g, based on a discount curve g, satisfies $\Pi_i(Y_i^g) = P_i^g = C_i g(\boldsymbol{x})$, consistent with the discounted cash flow equation (1). Using a first-order approximation, $P_i^g - P_i \approx \Pi_i'(Y_i)(Y_i^g - Y_i)$, we express the squared YTM error as an approximately weighted squared price error,

$$(Y_i^g - Y_i)^2 \approx \frac{1}{\left(\Pi_i'(Y_i)\right)^2} (P_i^g - P_i)^2.$$

YTM is often used to compare fixed-income instruments across maturities. Weighting squared price errors by $\omega_i = \frac{1}{M} \frac{1}{\left(\Pi_i'(Y_i)\right)^2}$ therefore ensures that estimation errors are more uniformly comparable across the maturity spectrum.

This logic also applies to swaps, for which either $P_i = 1$ (spot-starting) or $P_i = 0$ (forward-starting). The following result links YTM to the swap rate R:

Lemma 6.1. *If $T_j - T_{j-1} \equiv \Delta$ for all $j = 1, \ldots, n$, then $\Delta Y = \log(1 + \Delta R)$. That is, in first order the YTM equals the swap rate, $Y \approx R$.*

The following example illustrates this for a single-period overnight swap.

Example 6.2. In the U.S., the overnight RFR is SOFR, here denoted by R_{SOFR}. Consider a single-period overnight swap maturing at $T_1 = \frac{1}{365}$. The pricing equation (11) here is $P^g_{\text{SOFR}} = \left(1 + T_1 R_{\text{SOFR}}\right) g(T_1) - 1$.

The corresponding YTM Y_{SOFR} satisfies

$$0 = \left(1 + T_1 R_{\text{SOFR}}\right) e^{-Y_{\text{SOFR}} T_1} - 1,$$

which implies

$$Y_{\text{SOFR}} = \frac{1}{T_1} \log(1 + T_1 R_{\text{SOFR}}) \approx R_{\text{SOFR}}.$$

The derivative of the YTM–price function is $\Pi'_{\text{SOFR}}(Y_{\text{SOFR}}) = -T_1$, so the corresponding duration-based weight becomes $\omega_{\text{SOFR}} = \frac{1}{MT_1^2}$.

We use the separable kernel $K(x, y) = Bk(x, y)$ from Theorem 4.1 with scalar kernel given by

$$k(x,y) = -\frac{\min\{x,y\}}{\alpha^2} \mathrm{e}^{-\alpha \min\{x,y\}} + \frac{2}{\alpha^3}\left(1 - \mathrm{e}^{-\alpha \min\{x,y\}}\right) - \frac{\min\{x,y\}}{\alpha^2} \mathrm{e}^{-\alpha \max\{x,y\}} \tag{14}$$

for a maturity weight parameter $\alpha > 0$, which is introduced in [FPY24].[10] They show that the corresponding RKHS $\mathcal{H}_k$ is the weighted Sobolev space consisting of twice weakly differentiable functions $h : [0, \infty) \to \mathbb{R}$ with $h(0) = 0$, $\lim_{x\to\infty} h'(x) = 0$, and finite smoothness norm given by

$$\|h\|^2_{\mathcal{H}_k} = \int_0^\infty h''(x)^2 \mathrm{e}^{\alpha x}\, dx. \tag{15}$$

In sum, this specification includes the following hyperparameters: α (scalar kernel parameter), γ_a (discount curve smoothness), Θ_{ab} (spread smoothness). To reduce complexity, we set $\gamma_a = \gamma$ for all a and $\Theta_{ab} = \theta$ for all $a < b$.

We set $\alpha = 0.05$ and $\gamma = 10^{-4}$, as calibrated in [FPY24] for U.S. data.[11] The remaining tuning parameter is θ, which governs the strength of transfer learning. We consider values $\theta \in \{0, 1, 10, 100\}$. Setting $\theta = 0$ corresponds to independent estimation without transfer learning.

6.3 Illustrative effects of transfer learning

To demonstrate the effects of transfer learning, we present two representative examples in the main text: EUR and USD market, both for the date 2024-06-14. For size reasons, we show only these two instances here. The full set of results across all currencies and dates is provided in the supplementary online appendix.[12]

Figure 2 shows yield curves (left column) and forward curves (right column) for the EUR market on 2024-06-14, estimated under varying values of the transfer learning parameter θ.[13] Shaded areas represent 3σ confidence bands derived from the Gaussian process view of the KR estimator, cut at $\pm 2\%$ for readability.[14] German government bond maturities extend to 30 years, while ESTR swap tenors reach up to 60 years. Transfer learning leverages this longer maturity coverage to improve extrapolation for the bond yield curve.

[10] The RKHS introduced in [FPY24] is more flexible, as its norm (15) includes both first- and second-order derivatives. However, their empirical analysis on U.S. data finds that only the second-order term is relevant for the performance of the KR estimator. [CF24] confirm this finding for Swiss data as well.

[11] In fact, [FPY24] use $\gamma = 1/(365 \cdot x_N)$ for the prevailing last maturity date x_N, although the differences are economically negligible. [CF24] find that $\alpha = 0.02$ and $\gamma = 10^{-3}$ are statistically optimal for Swiss data. However, they also show that variations in α and γ within this range have no economically significant impact on the KR estimation.

[12] Available at Online Appendix for Transfer Learning Across Fixed-Income Product Classes.

[13] The yield curve consists of the annualized zero-coupon log returns, $y(x) = -\frac{1}{x} \log g(x)$, and the forward curve is defined as the logarithmic derivative $f(x) = -\frac{d}{dx} \log g(x)$, for a scalar-valued discount curve $g(x)$.

[14] Confidence bounds for the discount curve are computed as $\bar{g}_{a,\text{low}}(x) = \max(\bar{g}_a(x) - 3\sigma_a(x), \bar{g}_a(x)e^{-0.02x})$ and $\bar{g}_{a,\text{up}}(x) = \min(\bar{g}_a(x) + 3\sigma_a(x), \bar{g}_a(x)e^{0.02x})$, where $\sigma_a(x) = (\hat{s}K^{\text{post}}_{aa}(x, x))^{1/2}$. Confidence bands are not available for forward curves.

As θ increases, the confidence band for the bond yield curve narrows beyond 30 years, reflecting reduced uncertainty. In contrast, the swap yield curve, supported by denser data, remains largely unchanged. Within the interpolation range (up to 30 years), the bond yield curve is unaffected, confirming that transfer learning acts locally and preserves robustness in data-rich regions. As swaps typically span a broader maturity spectrum, information transfer tends to flow from swaps to bonds. The effects are even more apparent in the forward curves. For $\theta = 0$, the bond forward curve is visibly less smooth than the swap curve. For moderate values ($\theta = 1$ and 10), the swap forward curve imparts a smoothing effect on the bond forward curve. However, for large values ($\theta = 100$), the direction of influence reverses, and the bond forward curve begins to distort the swap forward curve, illustrating the risk of excessive transfer.

To quantify the transfer learning effects, Table 1 reports the change in yield $\Delta y = y_{x_N}(\theta > 0) - y_{x_N}(\theta = 0)$ at the longest available maturity x_N for each currency at each date. Across all five reference dates, changes for the swap yield curve remain negligible (below 1 basis point), whereas the bond yield curve exhibits economically meaningful shifts ranging from -26.3 to 22.0 basis points.[15]

Table 1: Effect of transfer learning on EUR yield curves (in basis points)

Date	$\theta = 1$		$\theta = 10$		$\theta = 100$	
	Swap	**Bond**	**Swap**	**Bond**	**Swap**	**Bond**
2020-06-15	0.3	-8.5	0.4	-18.6	0.4	-26.3
2021-06-15	-0.1	1.8	-0.3	0.6	-0.3	-3.5
2022-06-15	0.2	6.2	0.2	5.2	0.1	-2.1
2023-06-15	-0.4	8.6	-0.7	1.0	-0.9	-8.5
2024-06-14	-0.4	13.9	-0.7	22.0	-0.8	20.8

Figure 3 and Table 2 provide the corresponding results for the USD market on the same date. The patterns closely mirror those observed in the EUR case, reinforcing the conclusions above.

Table 2: Effect of transfer learning on USD yield curves (in basis points)

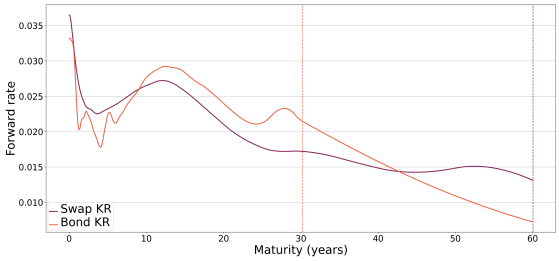
Date	$\theta = 1$		$\theta = 10$		$\theta = 100$	
	Swap	**Bond**	**Swap**	**Bond**	**Swap**	**Bond**
2020-06-15	0.1	4.8	0.1	12.3	-0.3	-2.5
2021-06-15	-0.1	3.0	-0.2	12.1	-0.4	12.8
2022-06-15	0.3	-4.8	0.6	13.3	0.4	25.2
2023-06-15	0.1	-3.0	0.2	14.5	0.1	33.5
2024-06-14	0.1	-3.8	0.1	12.0	0.0	34.0

These two examples are chosen for illustration due to their economic relevance and visual clarity. The full set of results, provided in the supplementary online appendix, consistently confirms the same qualitative pattern across all currencies and reference dates. In each case, transfer learning improves the estimation in data-scarce regions, particularly for government bond curves, while preserving robustness in well-identified segments.

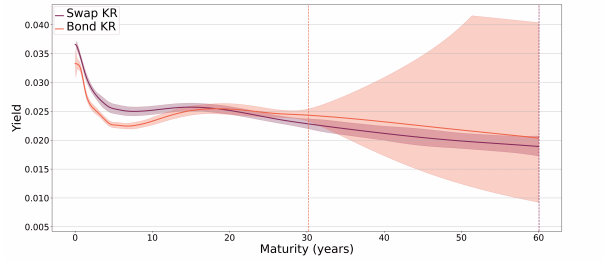
[15] For context, [FPY24] and [CF24] report YTM root mean squared errors (RMSE) between 5 and 8 basis points.

Figure 2: Transfer learning across government bonds and swaps: EUR yield and forward curves

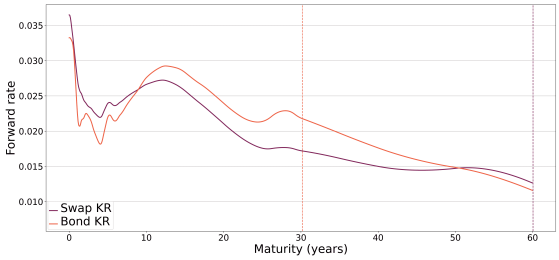
(a) $\theta = 0$



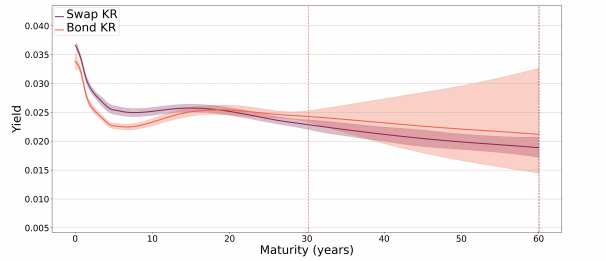
(b) $\theta = 0$



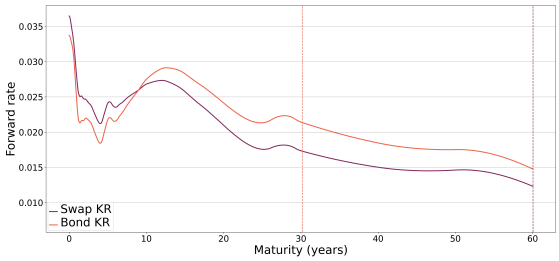
(c) $\theta = 1$



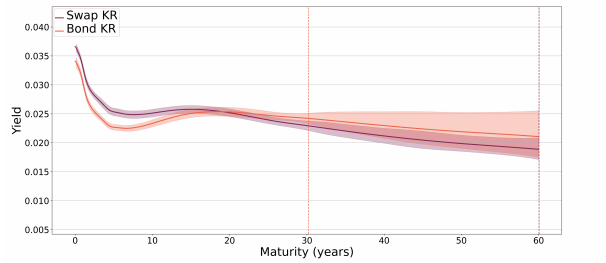
(d) $\theta = 1$



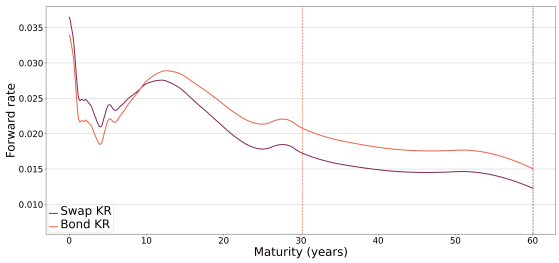
(e) $\theta = 10$



(f) $\theta = 10$



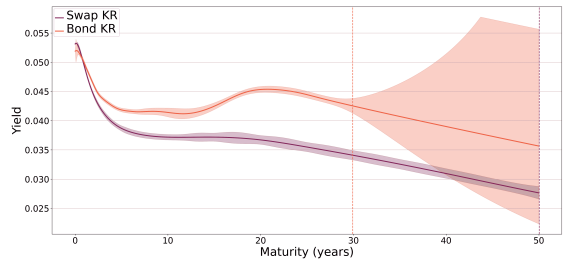
(g) $\theta = 100$



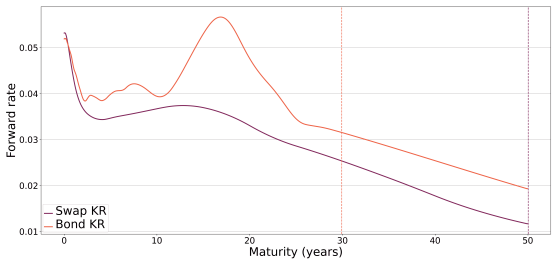
(h) $\theta = 100$

This figure displays yield curves (left column) and forward curves (right column) estimated on 2024-06-14 for German government bonds (Bond KR) and ESTR swaps (Swap KR), under varying values of the transfer learning parameter θ. In all panels, the vertical dashed lines indicate the longest available data point in the respective product class. The shaded areas show the 3σ confidence bands derived from the Gaussian process view and are capped at $\pm 2\%$.

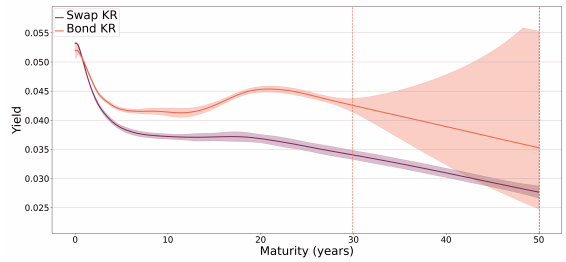
Figure 3: Transfer learning across government bonds and swaps: USD yield and forward curves



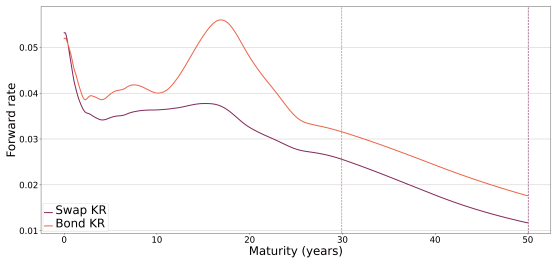
(a) $\theta = 0$



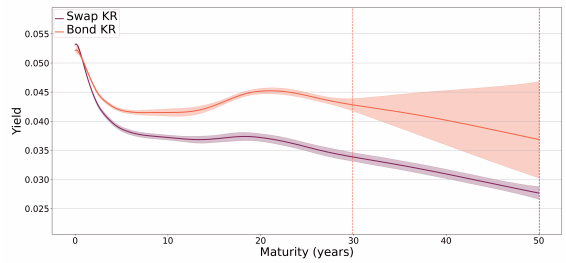
(b) $\theta = 0$



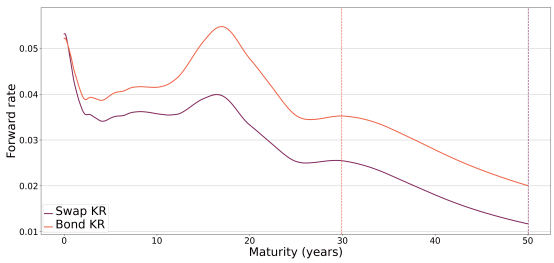
(c) $\theta = 1$



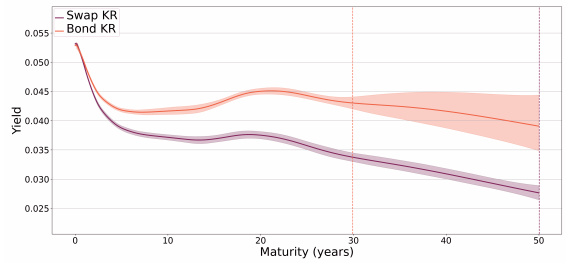
(d) $\theta = 1$



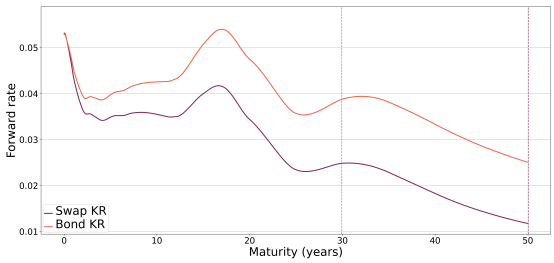
(e) $\theta = 10$



(f) $\theta = 10$



(g) $\theta = 100$



(h) $\theta = 100$

This figure displays yield curves (left column) and forward curves (right column) estimated on 2024-06-14 for U.S. government bonds (Bond KR) and SOFR swaps (Swap KR), under varying values of the transfer learning parameter θ. In all panels, the vertical dashed lines indicate the longest available data point in the respective product class. The shaded areas show the 3σ confidence bands derived from the Gaussian process view and are capped at $\pm 2\%$.

7 Conclusion

We introduce a transfer learning framework for jointly estimating discount curves across fixed-income product classes. Building on the discounted cash flow principle, our approach extends kernel ridge regression to a vector-valued setting, resulting in a convex optimization problem with a closed-form solution in a vector-valued RKHS. A key feature is the use of separable operator-valued kernels, which enable regularization of curve spreads in an economically meaningful way.

We derive a norm decomposition for separable kernels, generalizing prior results and leading to a principled spread regularization term. The framework admits a Gaussian process interpretation, allowing uncertainty quantification in the vector-valued setting.

We show how standard fixed-income instruments, including coupon bonds, interest rate swaps, and cross-currency swaps, can be embedded in this structure. Empirical illustrations across CHF, EUR, GBP, and USD markets demonstrate that transfer learning improves extrapolation while leaving well-identified regions unaffected. A comprehensive empirical assessment is left for future work.

References

[ARL12] Mauricio A. Alvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012. 2, 21

[BHJ+19] Thomas Brophy, Niko Herrala, Raquel Jurado, Irene Katsalirou, Léa Le Quéau, Christian Lizarazo, and Seamus O'Donnell. Role of cross currency swap markets in funding and investment decisions. Occasional Paper Series 228, European Central Bank, August 2019. 10

[Bjo09] Tomas Bjork. *Arbitrage Theory in Continuous Time*. Number 9780199574742 in OUP Catalogue. Oxford University Press, 2009. 26

[BoEa] BoE. SONIA key features and policies. https://www.bankofengland.co.uk/markets/sonia-benchmark/sonia-key-features-and-policies (accessed: 08.05.2025). 2

[BoEb] BoE. Sterling Overnight Index Average (SONIA). https://www.bankofengland.co.uk/markets/sonia-benchmark (accessed: 08.05.2025). 12

[BRBV12] Luca Baldassarre, Lorenzo Rosasco, Annalisa Barla, and Alessandro Verri. Multi-output learning via spectral filtering. *Machine Learning*, 87, 2012. 2, 8, 23

[Car97] Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997. 2

[CF24] Nicolas Camenzind and Damir Filipović. Stripping the swiss discount curve using kernel ridge regression. *European Actuarial Journal*, 14(2):371–410, June 2024. 1, 11, 12, 13, 14

[Chr17] Chris Barnes, Clarus FT. Mechanics of cross currency swaps, 2017. https://www.clarusft.com/mechanics-of-cross-currency-swaps/ (accessed: 08.05.2025). 28

[CWG19] Zexun Chen, Bo Wang, and Alexander N. Gorban. Multivariate gaussian and student-t process regression for multi-output prediction. *Neural Computing and Applications*, 32(8):3005–3028, dec 2019. 2, 7

[ECB] ECB. Euro short-term rate (€STR). https://www.ecb.europa.eu/stats/financial_markets_and_interest_rates/euro_short-term_rate/html/index.de.html (accessed: 08.05.2025). 2, 12

[FB87] Eugene F. Fama and Robert R. Bliss. The information in long-maturity forward rates. *The American Economic Review*, 77(4):680–692, 1987. 1, 12

[Fed] Fed. Secured Overnight Financing Rate (SOFR). https://www.newyorkfed.org/markets/reference-rates/sofr (accessed: 08.05.2025). 2, 12, 26

[FLT17] Damir Filipović, Martin Larsson, and Anders B Trolle. Linear-rational term structure models. *J. Finance*, 72(2):655–704, April 2017. 3

[FPY24] Damir Filipovic, Markus Pelger, and Ye Ye. Stripping the discount curve — a robust machine learning approach. *Management Science*, 2024. Accepted for publication. 1, 5, 6, 7, 8, 12, 13, 14, 24

[GSW07] Refet S. Gürkaynak, Brian Sack, and Jonathan H. Wright. The U.S. treasury yield curve: 1961 to the present. *Journal of Monetary Economics*, 54(8):2291–2304, 2007. 1, 12

[HJ12] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2 edition, 2012. 25

[Int20] International Capital Market Association. A quick guide to the transition to risk-free rates in the international bond market, 2020. pdf (accessed: 08.05.2025). 27

[JT95] Robert A Jarrow and Stuart M Turnbull. Pricing derivatives on financial securities subject to credit risk. *J. Finance*, 50(1):53, March 1995. 26

[Kat95] Tosio Kato. *Perturbation theory for linear operators*. Classics in Mathematics. Springer-Verlag, Berlin, 1995. Reprint of the 1980 edition. 22

[KDP+16] Hachem Kadri, Emmanuel Duflos, Philippe Preux, Stéphane Canu, Alain Rakotomamonjy, and Julien Audiffren. Operator-valued kernels for learning from functional response data. *Journal of Machine Learning Research*, 17(20):1–54, 2016. 2

[LW21] Yan Liu and Jing Cynthia Wu. Reconstructing the yield curve. *Journal of Financial Economics*, 142(3):1395–1425, 2021. 1, 12

[MM] Andréa M. Maechler and Thomas Moser. Life after Libor: A new era of reference interest rates. https://www.snb.ch/en/publications/communication/speeches/2022/ref_20220331_amrtmo (accessed: 08.05.2025). 2

[MP04] Charles A. Micchelli and Massimiliano Pontil. Kernels for multi-task learning. *NIPS*, 2004. 2

[MP05] Charles A. Micchelli and Massimiliano Pontil. On learning vector-valued functions. *Neural Computation*, 17, 2005. 2

[NS87] Charles R. Nelson and Andrew F. Siegel. Parsimonious modeling of yield curves. *The Journal of Business*, 60(4):473, January 1987. 1

[PR16] Vern I. Paulsen and Mrinal Raghupathi. *An introduction to the theory of reproducing kernel Hilbert spaces*, volume 152 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 2016. 2, 19, 20, 21

[PY10] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, October 2010. 2

[Ran23] Angelo Ranaldo. Foreign exchange swaps and cross-currency swaps. In Refet S. Gürkaynak and Jonathan H. Wright, editors, *Research Handbook of Financial Markets*, Chapters, chapter 20, pages 451–469. Edward Elgar Publishing, 2023. 10

[RW05] Carl Edward Rasmussen and Christopher K I Williams. *Gaussian processes for machine learning*. Adaptive Computation and Machine Learning series. MIT Press, London, England, November 2005. 2

[She08] Daniel Sheldon. Graphical multi-task learning. Technical report, Cornell University, 2008. 2, 8

[SIX] SIX. Swiss Reference Rates (SARON). https://www.six-group.com/en/market-data/indices/switzerland/saron.html (accessed: 08.05.2025). 2, 12, 26

[SK03] Alexander J. Smola and Risi Kondor. *Kernels and Regularization on Graphs*, page 144–158. Springer Berlin Heidelberg, 2003. 2

[SS19] Andreas Schrimpf and Vladyslav Sushko. Beyond libor: a primer on the new benchmark rates. *BIS Quarterly Review*, 2019. 2, 12

[Sve94] Lars Svensson. Estimating and interpreting forward interest rates: Sweden 1992 - 1994. NBER Working Papers 4871, National Bureau of Economic Research, Inc, 1994. 1

[SW01] Andrew Smith and Tim Wilson. Fitting yield curves with long term constraints. *Working paper*, 2001. 1

[Tea21] Risk.net Editorial Team. Beyond libor: the impact of sofr on rates, bonds and loans, 2021. https://www.risk.net/insight/markets/7957455/beyond-libor-the-impact-of-sofr-on-rates-bonds-and-loans (accessed: 08.05.2025). 27

[Wik] Wikipedia contributors. Kernel methods for vector output. https://en.wikipedia.org/wiki/Kernel_methods_for_vector_output (accessed: 08.05.2025). 21

[WJ24] David Wu and Robert A. Jarrow. The Treasury–SOFR swap spread puzzle explained, 2024. Available at SSRN: https://ssrn.com/abstract=4904777. 2, 27

[WKW16] Karl Weiss, Taghi M Khoshgoftaar, and Dingding Wang. A survey of transfer learning. *J. Big Data*, 3(1), December 2016. 2

A Vector-valued reproducing kernel Hilbert spaces

This appendix presents the theoretical background on vector-valued RKHS that underpins our transfer learning framework. For completeness, we begin by recalling the definition and main properties of vector-valued RKHS, following [PR16, Chapter 6]. Let E be any set and $A \in \mathbb{N}$.

Definition A.1. *An $\mathbb{R}^A$-valued* RKHS *on E is a Hilbert space $\mathcal{H}$ consisting of functions $h = (h_1, \dots, h_A)^\top : E \to \mathbb{R}^A$ such that for every $x \in E$, the linear evaluation map $E_x : \mathcal{H} \to \mathbb{R}^A$ given by $E_x(h) = h(x)$ is bounded.*

An $\mathbb{R}^A$-valued RKHS $\mathcal{H}$ has a reproducing kernel function $K : E \times E \to \mathbb{R}^{A \times A}$ defined by $K(x, y) = E_x E_y^*$, where we identify a linear operator on $\mathbb{R}^A$ with its $A \times A$-matrix representation in the standard Euclidean basis of $\mathbb{R}^A$. E_y^* denotes the adjoint operator. We immediately obtain that $K(\cdot, y)v = E_y^* v \in \mathcal{H}$ and $\langle K(\cdot, y)v, h\rangle_{\mathcal{H}} = v^\top h(y)$, for any $y \in E$, $v \in \mathbb{R}^A$, $h \in \mathcal{H}$. Moreover, we see that K is symmetric in the following sense,

$$K(x, y)^\top = K(y, x). \tag{16}$$

Note, however, that the matrices $K(x, y)$ are not symmetric for $x \neq y$ in general.[16] Moreover, for any finite points $x_1, \dots, x_n \in E$ the operator $(K(x_i, x_j))$ on $(\mathbb{R}^A)^A$ is positive semi-definite in the sense that for all choices of vectors $v_1, \dots, v_n \in \mathbb{R}^A$ we have

$$\sum_{i,j=1}^{n} v_i^\top K(x_i, x_j) v_j \geq 0. \tag{17}$$

Conversely, this leads to the following definition.[17]

Definition A.2. *A function $K : E \times E \to \mathbb{R}^{A \times A}$ satisfying (16) and (17) is called a* $\mathbb{R}^{A \times A}$-valued kernel function.

It follows by inspection that a function $K : E \times E \to \mathbb{R}^{A \times A}$ is a $\mathbb{R}^{A \times A}$-valued kernel function if and only if there exists a scalar kernel function k on $\{1, \dots, A\} \times E$ such that $K_{ab}(x, y) = k((a, x), (b, y))$.

Example A.3. The concept of matrix-valued kernels is surprisingly strong as it is somewhat difficult to generate examples easily. However, one possible way is to let $k_1, k_2, \dots, k_A$ be scalar kernels on E, then $K(x, y) = \operatorname{diag}(k_1(x, y), \dots, k_A(x, y))$ is a $\mathbb{R}^{A \times A}$-valued kernel. Indeed, property (16) holds because $K(x, y) = K(y, x)$ is symmetric. Property (17) is valid since

$$\sum_{i,j=1}^{n} v_i^\top K(x_i, x_j) v_j = \sum_{i,j=1}^{n} \sum_{a=1}^{A} v_{i,a} k_a(x_i, x_j) v_{j,a} = \sum_{a=1}^{A} \underbrace{\left(\sum_{i,j=1}^{n} v_{i,a} k_a(x_i, x_j) v_{j,a} \right)}_{\geq 0} \geq 0,$$

where the inner sums are non-negative due to the kernel property of each scalar kernel k_a.

Moore's vector-valued theorem [PR16, Theorem 6.12] states that for every $\mathbb{R}^{A \times A}$-valued kernel function K there exists a unique $\mathbb{R}^A$-valued RKHS $\mathcal{H}$ such that K is its reproducing kernel function. Moreover, functions of the form

$$h(x) = \sum_{j=1}^{n} K(x, y_j) v_j, \quad v_j \in \mathbb{R}^A, \quad y_1, \dots, y_n \in E, \quad n \in \mathbb{N}, \tag{18}$$

[16] Many papers in the literature assume that the matrices $K(x, y)$ are symmetric. But this is not the case in general, and, in fact, it excludes many examples.

[17] Note that [PR16, Definition 6.11] does not require (16) because they work on complex Hilbert spaces, where the non-negativity, that is, (17) with v_i replaced by its complex conjugate, already implies that $K(x, y)^* = K(y, x)$.

are dense in $\mathcal{H}$, see [PR16, Proposition 6.7].[18] A special class of $\mathbb{R}^{A\times A}$-valued kernels are separable kernels. In fact, as it turns out they are tractable and easy to interpret.

Definition A.4. *A $\mathbb{R}^{A\times A}$-valued kernel K on E is separable if it can be written as $K(x,y) = Bk(x,y)$ for some $A\times A$-matrix B and a scalar kernel k on E. In view of (16), the matrix B is necessarily symmetric positive semi-definite.*

Remark A.5. *Separable kernels are one of the simplest matrix-valued kernel. If we regard kernels as similarity measures, B encodes similarity across components while k encodes similarity across the space E.*

The following theorem provides an important representation result, which is at the heart of transfer learning in this paper.

Theorem A.6. *Let $\mathcal{H}$ be the vector-valued RKHS corresponding to the separable kernel $K(x,y) = Bk(x,y)$. Let $\mathcal{H}_k$ denote the RKHS corresponding to the scalar kernel k. Let Q be any generalized inverse $A\times A$-matrix of B such that $BQB = B$. Then the following hold.*

(i) *$\mathcal{H}$ is isomorphic to the direct sum $\bigoplus_{a=1}^{\tilde{A}} \mathcal{H}_k$, where $\tilde{A} = \operatorname{rank} B$.*

(ii) *$\mathcal{H} \subseteq (\mathcal{H}_k)^A = \mathcal{H}_k \times \cdots \times \mathcal{H}_k$ as sets, with equality if and only if B is non-singular.*

(iii) *For any $h = (h_1, \ldots, h_A)^\top \in \mathcal{H}$, the $\mathcal{H}$-norm can be expressed as*

$$\|h\|_{\mathcal{H}}^2 = \sum_{a,b=1}^{A} Q_{ab}\langle h_a, h_b\rangle_{\mathcal{H}_k}. \tag{19}$$

(iv) *If Q is symmetric then (19) can also be written as*

$$\|h\|_{\mathcal{H}}^2 = \sum_{a=1}^{A} \gamma_a \|h_a\|_{\mathcal{H}_k}^2 - \sum_{a=1}^{A}\sum_{b>a} Q_{ab}\|h_a - h_b\|_{\mathcal{H}_k}^2, \tag{20}$$

where $\gamma_a = \sum_{b=1}^{A} Q_{ab}$ denote the row sums.

Proof. We define the linear subspace $\mathcal{D}$ of $\mathcal{H}$ that consists of all functions of the form

$$h(\cdot) = \sum_{j=1}^{n} Bv_j k(\cdot, y_j), \quad v_j \in \mathbb{R}^A, \quad y_1, \ldots, y_n \in E, \quad n \in \mathbb{N}. \tag{21}$$

From (18) we know that $\mathcal{D}$ is dense in $\mathcal{H}$. Similarly, we define the dense subspace $\mathcal{D}_k$ of $\mathcal{H}_k$ of all functions of the form $g(\cdot) = \sum_{j=1}^{n} c_j k(\cdot, y_j)$, for $c_j \in \mathbb{R}$. Consequently, the direct sum $\bigoplus_{a=1}^{\tilde{A}} \mathcal{D}_k$ is a dense subspace of $\bigoplus_{a=1}^{\tilde{A}} \mathcal{H}_k$.

We prove the theorem in two steps. First, we prove all statements for $\mathcal{H}$ and $\mathcal{H}_k$ replaced by $\mathcal{D}$ and $\mathcal{D}_k$. Second, we argue by the continuous extension principle that all results carry over to $\mathcal{H}$ and $\mathcal{H}_k$.

We let $B = USU^\top$ denote the reduced spectral decomposition where U is an orthogonal $A\times\tilde{A}$-matrix such that $U^\top U = I_{\tilde{A}}$, and $S = \operatorname{diag}(s_1, \ldots, s_{\tilde{A}})$ contains the positive eigenvalues $s_1 \geq \cdots \geq s_{\tilde{A}} > 0$ of B.

We define the linear operator $\mathcal{U} : \mathcal{D} \to \bigoplus_{a=1}^{\tilde{A}} \mathcal{D}_k$, by $\mathcal{U}h(\cdot) = U^\top \sum_{j=1}^{n} Bv_j k(\cdot, y_j) = \sum_{j=1}^{n} SU^\top v_j k(\cdot, y_j)$. The operator $\mathcal{U}$ is injective, because $\mathcal{U}h(\cdot) = 0$ implies that $SU^\top v_j = 0$ and thus $v_j = 0$ for all $j = 1, \ldots, n$,

[18] Note that (18) differs from the corresponding formulas in [ARL12, page 209] and the wikipedia page [Wik]. The latter formulas are correct only if $K(x,y)$ is a symmetric matrix, which in view of (16) is not true in general.

hence $h = 0$. Here we assume that n is minimal in the sense that $k(\cdot, y_1), \dots, k(\cdot, y_n)$ are linearly independent in $\mathcal{H}_k$, without loss of generality. We claim that $\mathcal{U}$ is also surjective, $\mathcal{U}(\mathcal{D}) = \bigoplus_{a=1}^{\tilde{A}} \mathcal{D}_k$. Indeed, any $g \in \bigoplus_{a=1}^{\tilde{A}} \mathcal{D}_k$ can be written as $g(\cdot) = \sum_{j=1}^n w_j k(\cdot, y_j)$, for some $w_j \in \mathbb{R}^{\tilde{A}}$, $y_1, \dots, y_n \in E$, $n \in \mathbb{N}$. Define the linear operator $\mathcal{V} : \bigoplus_{a=1}^{\tilde{A}} \mathcal{D}_k \to \mathcal{D}$ by $\mathcal{V}g(\cdot) = U \sum_{j=1}^n w_j k(\cdot, y_j)$. As the $\tilde{A} \times A$-matrix $SU^\top$ has full rank $\tilde{A}$, there exist $v_j \in \mathbb{R}^A$ such that $w_j = SU^\top v_j$. Then $h \in \mathcal{D}$ given by $h(\cdot) = \mathcal{V}g(\cdot) = \sum_{j=1}^n Bv_j k(\cdot, y_j)$ is a pre-image of g, because $\mathcal{U}h(\cdot) = U^\top U \sum_{j=1}^n w_j k(\cdot, y_j) = g(\cdot)$. We conclude that $\mathcal{U} : \mathcal{D} \to \bigoplus_{a=1}^{\tilde{A}} \mathcal{D}_k$ is a linear bijection with inverse given by $\mathcal{U}^{-1} = \mathcal{V}$, which proves (i).

We also obtain that the components h_a of any $h \in \mathcal{D}$ are linear combinations of functions $g_b \in \mathcal{D}_k$ and thus elements in $\mathcal{D}_k$ themselves. As $\tilde{A} = A$ if and only if B is non-singular, this proves (ii).

Next we claim that (19) holds for $h \in \mathcal{D}$. Indeed, on one hand we have

$$\|h\|_{\mathcal{H}}^2 = \sum_{i,j=1}^n \langle Bv_i k(\cdot, y_i), Bv_j k(\cdot, y_j) \rangle_{\mathcal{H}} = \sum_{i,j=1}^n v_i^\top B v_j k(y_i, y_j)$$

by the basic reproducing kernel property of $K(\cdot, y_i) = Bk(\cdot, y_i)$. On the other hand, the right hand side of (19) equals

$$\sum_{a,b=1}^A Q_{ab} \langle h_a, h_b \rangle_{\mathcal{H}_k} = \sum_{i,j=1}^n \sum_{a,b=1}^A Q_{ab} (Bv_i)_a (Bv_j)_b k(y_i, y_j) = \sum_{i,j=1}^n v_i^\top BQB v_j k(y_i, y_j),$$

which equals the former and thus proves (iii).

As for (20), straightforward rearrangement of sums shows that the right hand side of (20) equals

$$\begin{aligned} RHS &= \sum_{a=1}^A Q_{aa} \|h_a\|_{\mathcal{H}_k}^2 + \sum_{a=1}^A \sum_{b \neq a} Q_{ab} \left(\|h_a\|_{\mathcal{H}_k}^2 - \frac{1}{2} \|h_a - h_b\|_{\mathcal{H}_k}^2 \right) \\ &= \sum_{a=1}^A Q_{aa} \|h_a\|_{\mathcal{H}_k}^2 + \sum_{a=1}^A \sum_{b \neq a} Q_{ab} \langle h_a, b_b \rangle_{\mathcal{H}_k} = \sum_{a,b=1}^A Q_{ab} \langle h_a, h_b \rangle_{\mathcal{H}_k}. \end{aligned}$$

In view of (19), this proves (iv).

We now extend the validity of the above proved properties to $\mathcal{H}$ and $\mathcal{H}_k$. Thereto, when writing $h = \mathcal{U}^{-1} g$ for $g = \mathcal{U}h \in \bigoplus_{a=1}^{\tilde{A}} \mathcal{D}_k$, we observe that the right hand side of (19) becomes

$$\|h\|_{\mathcal{H}}^2 = \sum_{a=1}^{\tilde{A}} s_a^{-1} \|g_a\|_{\mathcal{H}_k}^2. \tag{22}$$

Indeed, $BQB = B$ implies $U^\top QU = S^{-1}$, and thus $v^\top Qv = w^\top S^{-1} w$ for any $v = Uw$, which shows (22).[19] We obtain the bounds $s_1^{-1} \|g\|_{\bigoplus_{a=1}^{\tilde{A}} \mathcal{H}_k}^2 \leq \|h\|_{\mathcal{H}}^2 \leq s_{\tilde{A}}^{-1} \|g\|_{\bigoplus_{a=1}^{\tilde{A}} \mathcal{H}_k}^2$

We infer that $\mathcal{U} : \mathcal{D} \subset \mathcal{H} \to \bigoplus_{a=1}^{\tilde{A}} \mathcal{H}_k$ is bounded with operator norm $\|\mathcal{U}\| = s_1$. In the same vein, $\mathcal{U}^{-1} : \bigoplus_{a=1}^{\tilde{A}} \mathcal{D}_k \subset \bigoplus_{a=1}^{\tilde{A}} \mathcal{H}_k \to \mathcal{H}$ is bounded with operator norm $\|\mathcal{U}^{-1}\| = s_{\tilde{A}}^{-1}$. By the extension principle for bounded densely defined operators on Banach spaces, [Kat95, Section III.2.2], $\mathcal{U}$ uniquely extends to an invertible bounded operator $\mathcal{U} : \mathcal{H} \to \bigoplus_{a=1}^{\tilde{A}} \mathcal{H}_k$ with inverse given by the respective extension of $\mathcal{U}^{-1}$. As norm convergence in $\mathcal{H}$ and $\bigoplus_{a=1}^{\tilde{A}} \mathcal{H}_k$ implies point-wise convergence, we have $\mathcal{U}h(\cdot) = U^\top h(\cdot)$ and

[19] In more detail: we have $h_a = \sum_{i=1}^{\tilde{A}} U_{ai} g_i$, and hence $\sum_{a,b=1}^A Q_{ab} \langle h_a, h_b \rangle_{\mathcal{H}_k} = \sum_{i,j=1}^{\tilde{A}} \sum_{a,b=1}^A U_{ai} Q_{ab} U_{bj} \langle g_i, g_j \rangle_{\mathcal{H}_k}$, which equals the right hand side of (22).

$\mathcal{U}^{-1}g(\cdot) = Ug(\cdot)$, for all $h \in \mathcal{H}$ and $g \in \bigoplus_{a=1}^{\tilde{A}} \mathcal{H}_k$. The validity of (i), (ii), (iii), (iv) for $\mathcal{H}$ and $\mathcal{H}_k$ now follows by continuity arguments. □

Remark A.7. *Equation (19) is also proved in [BRBV12, Proposition 1], however, only for simple functions of the form (21), which corresponds to the first step in our proof of Theorem A.6.*

The following two auxiliary lemmas are of independent interest and potentially useful for the specification of a matrix-valued kernel. They provide general elementary decomposition results, which are known as kernel normalization in the scalar case.

Lemma A.8. *Any $\mathbb{R}^{A\times A}$-valued kernel K can be decomposed in the following way*

$$K(x,y) = S(x)R(x,y)S(y) \tag{23}$$

where R is a normalized $\mathbb{R}^{A\times A}$-valued kernel such that $R_{aa}(x,x) = 1$, and $S(x)$ is a diagonal matrix with non-negative elements, for all $a = 1,\dots,A$ and $x \in E$.

A particular decomposition is given by

$$S_{aa}(x) = K_{aa}(x,x)^{\frac{1}{2}} \tag{24}$$

and

$$R_{ab}(x,y) = \begin{cases} 1, & \text{if } a = b \text{ and } x = y, \\ S_{aa}(x)^{-1}K_{ab}(x,y)S_{bb}(y)^{-1}, & \text{if } S_{aa}(x) > 0 \text{ and } S_{bb}(y) > 0, \end{cases} \tag{25}$$

and we set

$$R_{ab}(x,y) = 0 \quad \text{otherwise.} \tag{26}$$

On the other hand, any such decomposition necessarily satisfies (24) and (25).[20]

Proof. Necessity of (24) and (25) follows by inspection.

It remains to prove that $R_{ab}(x,y)$ given by (25) and (26) defines a $\mathbb{R}^{A\times A}$-valued kernel. It is readily verified that $R_{ab}(x,y) = R_{ba}(y,x)$, which proves (16). As for (17), we define the index set $\mathcal{I}_0 = \{(a,i) \mid S_{aa}(x_i) = 0\}$ and its complement $\mathcal{I}_1 = \mathcal{I}_0^c$. Now let $v_1,\dots,v_n \in \mathbb{R}^A$, and define $w_i \in \mathbb{R}^A$ by $w_{ia} = v_{ia}S_{aa}(x_i)^{-1}$ for $(a,i) \in \mathcal{I}_1$ and $w_{ia} = 0$ otherwise. Then we have

$$\begin{aligned} \sum_{i,j=1}^{n} v_i^\top R(x_i,x_j)v_j = \sum_{a,b=1}^{A}\sum_{i,j=1}^{n} v_{ia}R_{ab}(x_i,x_j)v_{jb} &= \sum_{(a,i)\in\mathcal{I}_0} v_{ia}^2 + \sum_{(a,i),(b,j)\in\mathcal{I}_1} v_{ia}R_{ab}(x_i,x_j)v_{jb} \\ &\geq \sum_{(a,i),(b,j)\in\mathcal{I}_1} w_{ia}K_{ab}(x_i,x_j)w_{jb} = \sum_{i,j=1}^{n} w_i^\top K(x_i,x_j)w_j \geq 0 \end{aligned}$$

by the kernel property (17) of K. This completes the proof. □

In the special case of separable kernels, Lemma A.8 extends as follows.

[20] Property (26) does not necessarily hold. Indeed, consider the finite set $E = \{x_1,x_2\}$ and $A = 1$ and suppose that $K(x_1,x_1) = 1$ and $K(x_1,x_2) = K(x_2,x_2) = 0$. Then $R(x_1,x_1) = 1$, $R(x_1,x_2) = 1/2$ and $R(x_2,x_2) = 1$ is a normalized kernel satisfying the decomposition (23), but not (26).

Lemma A.9. *Let $K(x,y) = Bk(x,y)$ be a separable kernel. Then the normalized kernel given by (25) and (26) is separable of the form $R(x,y) = C\rho(x,y)$ for the symmetric positive semi-definite matrix C given by*

$$C_{ab} = \begin{cases} 1, & \text{if } a = b, \\ B_{aa}^{-\frac{1}{2}} B_{ab} B_{bb}^{-\frac{1}{2}}, & \text{if } B_{aa} > 0 \text{ and } B_{bb} > 0, \end{cases}$$

and we set $C_{ab} = 0$ otherwise and the scalar kernel ρ given by

$$\rho(x,y) = \begin{cases} 1, & \text{if } x = y, \\ k(x,x)^{-\frac{1}{2}} k(x,y) k(y,y)^{-\frac{1}{2}}, & \text{if } k(x,x) > 0 \text{ and } k(y,y) > 0, \end{cases}$$

and we set $\rho(x,y) = 0$ otherwise. In particular, C and ρ are normalized in the sense that $C_{aa} = 1$ and $\rho(x,x) = 1$, for all $a = 1, \dots, A$ and $x \in E$.

Proof. It is enough to show that C is a symmetric positive semi-definite matrix and ρ a scalar kernel. This can both be proved using similar arguments as in the proof of Lemma A.8. □

B Proofs

This appendix provides the proofs of the results stated in the main text, based on the foundational material presented in Appendix A.

B.1 Proof of Theorem 2.1

Let S be the sampling operator as in equation (27). For any $m \in \{1, \dots, M\}$ define $a(m), i(m)$ such that $\boldsymbol{C}_m = (\dots, C_{a(m),i(m)}, \dots)$ is the m-th row of $\boldsymbol{C}$, and $\boldsymbol{P}_m = P_{a(m),i(m)}$ is the m-th component of $\boldsymbol{P}$, and $\boldsymbol{\omega}_m = \omega_{a(m),i(m)}$ the corresponding weight. Then the weighted mean-squared pricing error can be written as

$$\sum_{m=1}^{M} \boldsymbol{\omega}_m (\boldsymbol{P}_m - \boldsymbol{C}_m \operatorname{vec}(p^\top(\boldsymbol{x})) - \boldsymbol{C}_m S h)^2.$$

Similarly for the constraints, where $\boldsymbol{\omega}_m = \infty$.

It then follows that the solution of the KR problem must lie in the orthogonal complement of the null space of $\boldsymbol{C}S$. That is, $h = S^* \boldsymbol{C}^\top q$, for some $q \in \mathbb{R}^M$. The rest of the proof now follows as in the scalar case [FPY24, Theorem A.1], using Lemma B.1 below. This completes the proof of Theorem 2.1.

Lemma B.1. *Define the sampling operator $S : \mathcal{H} \to \mathbb{R}^{AN}$ by*

$$Sh = \operatorname{vec}(h^\top(\boldsymbol{x})). \tag{27}$$

The adjoint $S^* : \mathbb{R}^{AN} \to \mathcal{H}$ is given by

$$S^* v = \sum_{j=1}^{N} K(\cdot, x_j) V_j^\top \tag{28}$$

where V_j is the j-th row of the matrix $V \in \mathbb{R}^{N \times A}$ with $\operatorname{vec}(V) = v$. Moreover, $\boldsymbol{K}$ is the matrix representation of the linear operator $SS^* : \mathbb{R}^{AN} \to \mathbb{R}^{AN}$ in the standard Euclidean basis of $\mathbb{R}^{AN}$.

Proof of Lemma B.1. Let $v \in \mathbb{R}^{AN}$ and $V \in \mathbb{R}^{N\times A}$ its matricization such that $\operatorname{vec}(V) = v$. Then

$$\langle Sh, v\rangle_{\mathbb{R}^{AN}} = \sum_{j=1}^{N}\sum_{a=1}^{A} h_a(x_j)V_{ja} = \sum_{j=1}^{N} V_j h(x_j) = \sum_{j=1}^{N}\langle h, K(\cdot, x_j)V_j^\top\rangle_{\mathcal{H}},$$

which proves (28). In coordinates, (28) reads as

$$S^* v = \sum_{b=1}^{A}\sum_{j=1}^{N} \left(K_{1b}(\cdot, x_j), K_{2b}(\cdot, x_j), \ldots, K_{Ab}(\cdot, x_j)\right)^\top V_{jb},$$

and thus we obtain

$$SS^* v = \sum_{b=1}^{A}\sum_{j=1}^{N} \operatorname{vec}\left(K_{1b}(\boldsymbol{x}, x_j), K_{2b}(\boldsymbol{x}, x_j), \ldots, K_{Ab}(\boldsymbol{x}, x_j)\right) V_{jb} = \boldsymbol{K}v,$$

as desired. □

B.2 Proof of Theorem 4.1

According to Theorem A.6(iv) it is enough to construct a symmetric positive definite matrix Q such that $\gamma_a = \sum_{b=1}^{A} Q_{ab}$ and $Q_{ab} = -\Theta_{ab}$ for $a < b$. Therefore, we parameterize Q by the $A(A-1)/2$ spread smoothness parameters $\Theta_{ab} \geq 0$, as defined in (10).

By construction, the matrix Q is strictly diagonally dominant, $Q_{aa} > \sum_{b\neq a} |Q_{ab}|$, for all a, and hence positive definite, see [HJ12, Theorem 6.1.10]. Hence $B = Q^{-1}$ is symmetric and positive definite leading to a valid separable kernel. Theorem A.6 implies that the norm of the vector-valued RKHS $\mathcal{H}$ with separable kernel $K(x, y) = Bk(x, y)$ is given by (9). Theorem A.6 also implies that the optimization problem (8) over the product space $(\mathcal{H}_k)^A$ is equivalent to the KR problem (4) with norm (9) for $\lambda = 1$.

Remark B.2. *The matrix Q in (10) is strictly diagonally dominant, by construction. This is sufficient for Q being positive definite. However, not every symmetric positive definite matrix is strictly diagonally dominant. An example is given by*

$$Q = \begin{pmatrix} 4 & q \\ q & 1 \end{pmatrix},$$

for any $1 < q < 2$. Indeed, the characteristic polynomial is $(4-\lambda)(1-\lambda) - q^2 = \lambda^2 - 5\lambda + 4 - q^2$. Hence the eigenvalues of Q are positive, $\lambda_{1,2} = \frac{5\pm\sqrt{25-4(4-q^2)}}{2} > 0$, and Q is positive definite. However, Q is not diagonally dominant, as $Q_{22} = 1 < q = Q_{21}$. In that sense, specification (9) is a special case of a vector-valued RKHS with separable kernel as discussed in Theorem A.6

B.3 Proof of Lemma 6.1

Under the assumption of the lemma, we have after multiplication with $\mathrm{e}^{T_0 Y}$

$$0 = \Delta R \sum_{j=1}^{n} \mathrm{e}^{-\Delta Y j} + \mathrm{e}^{-\Delta Y n} - 1 = \Delta R \frac{q}{1-q}(1-q^n) - (1-q^n),$$

where we write $q = \mathrm{e}^{-\Delta Y}$. Therefore $\Delta R = \frac{1-q}{q}$, which proves the claim.

C Arbitrage-free pricing framework

In this appendix, we place the discounted cash flow equation (1) within an arbitrage-free pricing framework, following standard principles of asset pricing theory (see, e.g., [Bjo09]).

Let $(\Omega, \mathcal{F}, \mathbb{Q})$ be a probability space equipped with a filtration $(\mathcal{F}_t)_{t\geq 0}$ representing the flow of market information. All processes are assumed to be adapted to this filtration. The pricing measure $\mathbb{Q}$ is risk-neutral with respect to a numeraire $B(t)$, interpreted as the money market account, satisfying $B(0) = 1$ and accruing at the overnight RFR. The present value at time 0 of an $\mathcal{F}_T$-measurable cash flow Z paid at time $T > 0$ is

$$PV_Z = \mathbb{E}_{\mathbb{Q}}\left[\frac{Z}{B(T)}\right] = \mathbb{E}_{\mathbb{Q}^T}[Z]\, g_0(T), \tag{29}$$

where $g_0(T) = \mathbb{E}_{\mathbb{Q}}\left[\frac{1}{B(T)}\right]$ is the price of a risk-free discount bond maturing at T, and $\mathbb{Q}^T$ denotes the T-forward measure defined via the Radon–Nikodym derivative $\frac{d\mathbb{Q}^T}{d\mathbb{Q}} = \frac{1}{g_0(T)\, B(T)}$.

C.1 Non-defaultable bonds

Bonds issued by highly rated sovereigns, such as U.S. Treasuries or German government bonds, are typically regarded as non-defaultable (or risk-free). The discounted cash flow equation (1) applies directly with $g_a = g_0$ for such a bond paying nominal coupons $c_1, \ldots, c_n$ at dates $0 < T_1 < \cdots < T_n$ and the notional of one at the maturity T_n.

C.2 Defaultable bonds

Defaultable (or credit-risky) bonds include corporate debt and sovereign debt issued by less creditworthy countries. These instruments generally trade at a spread over the risk-free curve to reflect credit risk. Let τ denote the default time (which is a stopping time). Under the widely used recovery-of-treasury assumption (see [JT95]), the cash flow at T_i is modeled as

$$Z_i = c_i\, \mathbf{1}_{\{\tau > T_i\}} + c_i\, \delta_i\, \mathbf{1}_{\{\tau \leq T_i\}},$$

where $\delta_i \in [0, 1)$ is a deterministic recovery rate. Applying (29), we obtain

$$PV_{Z_i} = \mathbb{E}_{\mathbb{Q}^T}[Z_i]\, g_0(T_i) = c_i\big(\mathbb{Q}^{T_i}[\tau > T_i] + \delta_i\, \mathbb{Q}^{T_i}[\tau \leq T_i]\big) g_0(T_i),$$

which motivates the effective discount factor

$$g_a(T_i) = \big(\mathbb{Q}^{T_i}[\tau > T_i] + \delta_i\, \mathbb{Q}^{T_i}[\tau \leq T_i]\big) g_0(T_i). \tag{30}$$

Defaultable bonds are typically grouped by credit rating. Assuming all bonds within a given rating class a share the same default distribution and recovery profile, the class admits a common discount curve $g_a(x)$, and the discounted cash flow equation (1) applies.

C.3 RFR-based swaps

An RFR-based swap is an interest rate swap whose floating leg is linked to the money market account $B(t)$, which accrues at the RFR, such as SOFR in the United States or SARON in Switzerland, see [Fed, SIX].

Under the no-arbitrage assumption, RFR-based swap contracts should be priced using the same discount curve g_0 as creditworthy government bonds denominated in the same currency. In practice, however, a swap–government bond spread is observed. This spread arises due to market frictions and regulatory effects, and lies outside the scope of our simple arbitrage-free pricing framework, see, e.g., [WJ24].

As for the fixed leg, let $T_0 < T_1 < \cdots < T_n$ denote the payment dates, with notional normalized to one. For a given annualized swap rate R, the fixed cash flow at time T_i is ΔR with $\Delta = T_i - T_{i-1}$. By (29), the present value of the fixed leg is

$$PV_{\text{fixed}} = \Delta R \sum_{i=1}^{n} g_0(T_i).$$

Let $T_0 = t_0 < \cdots < t_m = T_n$ denote the reset and payment dates of the RFR floating leg, again with notional normalized to one. The floating cash flow at time $t_i > 0$ corresponds to the simple return of the money market account over the accrual period $[t_{i-1}, t_i]$, given by $\frac{B(t_i)}{B(t_{i-1})} - 1$. Using (29) and observing the telescoping structure of the discounted cash flows, we obtain $\sum_{i=1}^{m} \frac{1}{B(t_i)}\left(\frac{B(t_i)}{B(t_{i-1})} - 1\right) = \frac{1}{B(T_0)} - \frac{1}{B(T_n)}$, from which the present value of the RFR floating leg follows as

$$PV_{\text{RFR–floating}} = g_0(T_0) - g_0(T_n). \tag{31}$$

Although the above specification, where floating cash flows are "fixed in arrears," has become the standard, see, e.g., [Int20, Tea21], an alternative is to define the floating rate over $[t_{i-1}, t_i]$ as the simple return on a discount bond, $R_{\text{term}}(t_{i-1}, t_i) = \frac{1}{g_0(t_{i-1}, t_i)} - 1$. Here, with a slight abuse of notation, we denote by $g_0(t, T) = \mathbb{E}_{\mathbb{Q}}\left[\frac{1}{B(T)} \mid \mathcal{F}_t\right]$ the time-t value of a risk-free discount bond maturing at T, such that $g_0(x) = g_0(0, x)$. Under this alternative specification, the present value of the floating leg remains given by (31), which follows directly as a simple consequence of the arbitrage-free pricing formula (29).

C.4 IBOR swaps

Interest rate swaps whose floating leg is tied to an interbank loan term rate (IBOR) reflect credit and liquidity risk, which we model by adding a spread to the floating cash flows. For example, EURIBOR can be viewed as the sum of the risk-free ESTR and a credit spread capturing interbank risk.[21]

Formally, using the same tenor structures for the floating and fixed legs as in Subsection C.3, the floating cash flow of an IBOR swap at time t_i is given by $R_{\text{term}}(t_{i-1}, t_i) + S(t_{i-1}, t_i)$, where $S(t_{i-1}, t_i)$ denotes a spread that reflects the credit and liquidity risk of lending in the interbank market over the period $[t_{i-1}, t_i]$. The present value of the IBOR swap's floating leg is then

$$PV_{\text{IBOR–floating}} = g_0(T_0) - g_0(T_n) + \sum_{i=1}^{n} \mathbb{E}_{\mathbb{Q}^{t_i}}\left[S(t_{i-1}, t_i)\right] g_0(t_i). \tag{32}$$

As in the case of defaultable bonds discussed in Subsection C.2, we classify IBOR swaps according to the length of the accrual period (tenor) of the floating leg, such as quarterly, semiannual, or annual. We assume that all IBOR swaps within a given tenor class a share the same spread structure, which gives rise to a common discount curve $g_a(x)$. This curve is determined from the discounted cash flow equation (1), in conjunction with the expressions for the floating and fixed cash flows, resulting from (11) and (12), respectively.

[21] Strictly speaking, ESTR is not secured, unlike SOFR. However, as an overnight rate, its credit risk is considered negligible, and we treat it as risk-free for our purposes.

For positive spreads $S(t_{i-1}, t_i) > 0$, the discount curve implied by the IBOR swap is strictly below the RFR-based swap curve, that is, $g_a(x) < g_0(x)$. However, as seen from (32), this relationship is not as explicit as in the recovery-of-treasury model for defaultable bonds, as given in (30).

C.5 Cross-currency swaps

We are considering a standard floating–floating XCCY swap. The tenor structure of the cash flows is given by $0 \le t_0 < t_1 < \cdots < t_m$. The XCCY consists of two legs, leg a and leg b. Leg b is treated as the liquid leg. Thus, the basis spread s is added to leg a which has a normalized notional of 1. The initial notional of leg b is set to the spot exchange rate, $X_{ab}(t_0)$. The MTM feature is sometimes applied to leg b.

We now show that, when present, the MTM adjustments do not affect the present value of leg b. According to Clarus Financial Technology,[22] the floating cash flow Z_i at each payment date $t_i > 0$ consists of the simple return on the money market account applied to the MTM notional over the accrual period $[t_{i-1}, t_i]$, minus the change in MTM notionals over that period, and plus the MTM notional at maturity if $t_i = t_m$. Formally, this gives

$$Z_i = X_{ab}(t_{i-1})\left(\frac{B_b(t_i)}{B_b(t_{i-1})} - 1\right) - (X_{ab}(t_i) - X_{ab}(t_{i-1})) + X_{ab}(T)\,1_{t_i=t_m},$$

where $X_{ab}(t)$ denotes the MTM notional in currency b at time t, and $B_b(t)$ is the corresponding money market account.

Discounting each cash flow by the money market account and simplifying the telescoping sum yields

$$\sum_{i=1}^{m} \frac{Z_i}{B_b(t_i)} = \sum_{i=1}^{m} \left(\frac{X_{ab}(t_{i-1})}{B_b(t_{i-1})} - \frac{X_{ab}(t_i)}{B_b(t_i)}\right) + \frac{X_{ab}(t_m)}{B_b(t_m)} = X_{ab}(t_0),$$

which is known (deterministic) at time t_0 and equal to the initial notional. Hence, the present value of leg b is given by $X_{ab}(t_0)$, as in the case without MTM. This demonstrates that MTM adjustments, while relevant for risk management, do not affect the arbitrage-free valuation of the liquid leg.

[22] Clarus FT is a data and analytics provider focused on OTC derivatives markets. See [Chr17] for a discussion of MTM mechanics in cross-currency swaps.