Abstract. Distributionally robust optimization (DRO) is widely used, because it offers a way to overcome the conservativeness of robust optimization without requiring the specificity of stochastic optimization. On the computational side, many practical DRO instances can be equivalently (or approximately) formulated as semidefinite programming (SDP) problems via conic duality of the moment problem. However, despite being theoretically solvable in polynomial time, SDP problems in practice are computationally challenging and quickly become intractable with increasing problem size. In this paper, we propose a new approximation method to solve DRO problems with moment-based ambiguity sets. Our approximation method relies on principal component analysis (PCA) for optimal representation of variability in random samples. We show that the PCA approximation yields a relaxation of the original problem and derive theoretical bounds on the gap between the original problem and its PCA approximation. Furthermore, an extensive numerical study shows the strength of the proposed approximation method in terms of solution quality and runtime. For certain classes of distributionally robust conditional value-at-risk (CVaR) problems, the proposed PCA approximation using only 50% of the principal components provides a near-optimal solution (within 1%) with a one to two order of magnitude reduction in computation time. Finally, the proposed PCA approximation is exact when all principal components are used and significantly outperforms the original reformulation with respect to computation time.

Key words. stochastic programming, distributionally robust optimization, principal component analysis, semidefinite programming

AMS subject classifications. 90C15, 90C22, 90C59

1. Introduction. The ability of stochastic programming (SP) to incorporate uncertainty within an optimization framework is driving its increasing popularity, as it caters to the needs of modeling uncertain real-world problems. There are many applications of SP in energy, transportation, and finance [4, 7, 24]. We refer the reader to Prékopa [23] and Shapiro et al. [27] for details on the theory and applications of SP. In SP, the fundamental assumption is that the probability distributions of uncertain parameters are either known or can be estimated with some degree of accuracy. However, in many real-world applications this assumption is not necessarily realistic as the probability distribution is typically unknown as it may only be indirectly observable or estimated through limited historical samples. Due to the limited number of samples, estimation of the uncertainty space may be biased and subsequently the solution to the considered problem may be suboptimal.

A practical alternative to SP is robust optimization (RO), where only the support of uncertain parameters is assumed to be available. RO ensures that the solution is feasible for all realizations of the uncertain parameters, thus solutions of RO are robust against the full variability of the uncertain parameters. RO has been applied in a broad range of areas, such as control theory and energy [3, 20]. For a comprehensive overview of robust optimization theory and applications, we refer the reader to [1, 2] and references therein. However, aside from the support, RO models do not exploit additional distributional details, such as any moment information, which may be available even when the probability distribution is not fully known. It is beneficial
to employ advanced RO methods that leverage available distributional information, aside from the support, to reduce conservativeness and improve solution quality.

Distributionally robust optimization (DRO), which dates back to the 1950s [25], was developed to fill this gap. In DRO, the probability distributions of uncertain parameters are assumed to belong to an ambiguity set, a family of distributions that share the same information. Since the introduction of DRO by Scarf [25], a number of ambiguity sets have been proposed and analyzed. Among these, three types have received significant attention: moment-based ambiguity sets, structural ambiguity sets and metric-based ambiguity sets. In moment-based ambiguity sets, it is assumed that all distributions in the distribution family share the same moment information [6, 8, 32, 33]. In structural ambiguity sets, distributions share the same structural properties, such as symmetry, unimodality, and monotonicity [15, 22, 29, 30]. Metric-based ambiguity sets are created by requiring that all distributions are close to a reference (or nominal) distributions within a prespecified probability distance. The reference distribution is usually estimated using sampled data. Several types of probability distance functions have been proposed: the Prohorov metric [10], the \( \phi \)-divergence [5, 19], and the Wasserstein metric [11, 12]. Leveraging conic duality for moment problems [18, 26] and developments of interior point algorithms for solving semidefinite programming (SDP) problems, many DRO problems can be reformulated equivalently (or approximately) as SDP problems.

Although SDP problems can be solved theoretically in polynomial time, significant computational hurdles remain in practice for large size problems. As such, new approximation methods for solving DROs that provide efficient tradeoffs between solution quality and computational tractability are needed. The size of matrices in the SDP formulations of DRO problems increases quadratically with the number of random variables involved. Thus one potential approximation approach would employ dimensionality reduction approaches to reduce the number of random variables under consideration. In this context, Principal Component Analysis (PCA) is an effective technique. PCA employs a linear combination of orthogonal eigenmodes to provide an optimal representation of the variability in the data. For more information on PCA, we refer the interested reader to [31].

In this paper, we study DRO problems with moment-based ambiguity sets, which account for information about the support, mean and covariance of random variables. We present a dimensionality reduction scheme for DRO based on PCA, which allows for direct control of the trade-offs between solution quality and computation time.

The contributions of this paper can be summarized as follows.

1. We propose a new approximation method based on dimensionality reduction for DRO problems with moment-based ambiguity sets. More precisely, it is based on the idea of PCA. Additionally, the proposed approximation framework can also be extended to more general ambiguity sets.
2. We show that the PCA approximation yields a relaxation of the original problem and quantify the impact of the number of principal components on solution quality by deriving theoretical bounds on the gap between the original problem and its PCA approximation.
3. We prove that the PCA approximation becomes exact when the number of principal components is greater than or equal to the rank of covariance matrices. Further, it is shown that the PCA approximation is computationally more efficient in numerical tests due to the sparsity of the coefficients in its objective function as there are only \( m + 1 \) variables involved in the objective
function of the PCA approximation while it is $m^2 + m + 1$ for the original reformulation.

4. We demonstrate the efficacy of the proposed PCA approximation for solving large-scale problems and verify the theoretical results through a comprehensive numerical study on a distributionally robust conditional value-at-risk (CVaR) problem. For instance, in the numerical experiments, the proposed PCA approximation with half of the principal components yields a near-optimal solution (within 1%) with a one to two order of magnitude reduction in computation time.

The remainder of this paper is organized as follows. In Section 2, we present the DRO problem, develop the PCA approximation framework, and derive theoretical results. In Section 3, we extend the approximation method to distributionally robust chance-constrained programs. In Section 4, a comprehensive numerical study on a CVaR problem is conducted to demonstrate the strengths of the proposed approximation method. Finally, Section 5 discusses future work and concludes the paper.

2. PCA approximation for DRO problems. In this section, we describe the reformulation strategies of [8] to transform DRO problems with moment-based ambiguity sets into equivalent SDP problems. We then describe low-rank approximations for matrices and apply one such method, Principal Component Analysis, to effectively approximate the SDP reformulation of the DRO problem. Finally, we present theoretical bounds on the quality of the proposed PCA approximation.

2.1. SDP Reformulation for DRO problems. We consider a stochastic optimization problem with the following form:

$$\minimize_{x \in X} \mathbb{E}_F f(x, \xi)$$  \hspace{1cm} \text{(2.1)}$$

where $x \in \mathbb{R}^n$; $X \subset \mathbb{R}^n$ is a convex set; $\xi \in \mathbb{R}^m$ is random vector with a distribution $F$; and $f(x, \xi)$ is a cost function that is convex in $x$ given a fixed $\xi$. In stochastic optimization, it is typically assumed that distribution $F$ is known exactly. However, this assumption is overly restrictive in many cases. In many practical settings it is difficult to infer the exact distribution given limited data samples. In such cases, it may be necessary to work with only partial information on the distribution $F$, (e.g., its support and some moments). In other words, distribution $F$ belongs to some ambiguity set $\mathcal{D}$ that encompasses the partial information. Under a robust optimization framework, we can consider the worst-case result of the stochastic optimization problem, namely the distributionally robust optimization problem, as follows:

$$\minimize_{x \in X} \maximize_{F \in \mathcal{D}} \mathbb{E}_F f(x, \xi).$$  \hspace{1cm} \text{(2.2)}$$

Here (2.2) can be interpreted as a risk-averse (conservative) approximation of problem (2.1). In the remainder of the paper, we focus on the moment-based ambiguity set, where information on the support, mean and covariance of $\xi$ is known explicitly.

Assumption 2.1.1. The distributional ambiguity set, $\mathcal{D}(\mathcal{S}, \mu, \Sigma)$, accounts for information about the convex support $\mathcal{S}$, mean $\mu$ in the strict interior of $\mathcal{S}$, and an upper bound $\Sigma \succeq 0$ on the covariance matrix of the random vector $\xi$.

$$\mathcal{D}(\mathcal{S}, \mu, \Sigma) = \left\{ F \mid \begin{array}{l} \mathbb{P}(\xi \in \mathcal{S}) = 1 \\ \mathbb{E}_F[\xi] = \mu \\ \mathbb{E}_F[(\xi - \mu)(\xi - \mu)^T] \preceq \Sigma \end{array} \right\}. \hspace{1cm} \text{(2.3)}$$
\textbf{Remark 1.} An extension to a more general ambiguity set – for instance, the mean of \( \xi \) lies in an ellipsoid with the center \( \mu \) ([8]) – is straightforward and is omitted to simplify the introduction of the proposed approximation method.

Using ambiguity set \( D(S, \mu, \Sigma) \), Delage \textit{et al.} [8] showed that problem (2.2) can be reformulated as a semi-definite problem because of the conic duality for moment problems.

\textbf{Theorem 2.1.} Under assumption 2.1.1, if \( f(x, \xi) \) is \( F \)-integrable for any \( F \in D \), then problem (2.2) has the same optimal value as the following problem:

\begin{equation}
\min_{x, s, q, \mu} \ s + \mu^T q + (\Sigma + \mu\mu^T) \cdot Q
\end{equation}

subject to

\begin{equation}
\begin{aligned}
& s + \xi^T q + \xi^T Q \xi \geq f(x, \xi), \quad \forall \xi \in S \\
& Q \succeq 0, \ x \in X .
\end{aligned}
\end{equation}

where \( s \in \mathbb{R} \), \( q \in \mathbb{R}^m \), \( Q \in \mathbb{R}^{m \times m} \) and “\( \cdot \)” is the inner product defined by \( A \cdot B = \sum_{i,j} A_{ij} B_{ij} \) where \( A \) and \( B \) are two conformal matrices.

\textit{Proof.} The result can be directly deduced from Lemma 1 in [8]. \( \square \)

\textbf{Corollary 2.2.} When the support of \( F \) is polyhedral with at least one interior point, i.e., \( S = \{ \xi | A \xi \leq b \} \) with \( A \in \mathbb{R}^{n_1 \times m} \) and \( b \in \mathbb{R}^{n_1} \), if \( f(x, \xi) \) is a piecewise linear convex function in \( \xi \), precisely, \( f(x, \xi) = \max_{k=1}^K y_k^0(x) + y_k(x)^T \xi \) where \( y_k(x) = [y_k^0(x), \ldots, y_k^m(x)]^T \) as well as \( y_k^0(x) \) are affine in \( x \) for \( k = 1, \ldots, K \), problem (2.4) is reduced to the following problem

\begin{equation}
\min_{x, s, q, \mu} \ s + \mu^T q + (\Sigma + \mu\mu^T) \cdot Q
\end{equation}

subject to

\begin{equation}
\begin{aligned}
& \left[ s - y_k^0(x) - \lambda^T b \frac{(q - y_k(x) + A^T \lambda)^T}{2} \right] \geq 0, \forall k \in \{1, \ldots, K\} \\
& Q \succeq 0, \ \lambda \geq 0, \ x \in X ,
\end{aligned}
\end{equation}

where \( \lambda \in \mathbb{R}^{n_1} \).

\textit{Proof.} The basic idea of the proof is to apply the strong duality theorem to constraint (2.4b). Let

\[ Z_k = \left[ \begin{array}{c}
\frac{1}{2} s - y_k^0(x) - \lambda^T b \\
\frac{1}{2} (q - y_k(x) + A^T \lambda)^T \\
\end{array} \right] . \]

As \( f(x, \xi) \) is a piecewise linear convex function, constraint (2.4b) is reformulated as

\begin{equation}
s + \xi^T q + \xi^T Q \xi \geq y_k^0(x) + y_k(x)^T \xi, \quad \forall \xi \in S, \ \forall k \in \{1, 2, \ldots, K\} .
\end{equation}

Let \( g_k(\xi) := s + \xi^T q + \xi^T Q \xi - y_k^0(x) - y_k(x)^T \xi \), then constraint (2.6) is equivalent to minimize \( \inf_{A \xi \leq b, \xi \in \mathbb{R}^m} g_k(\xi) \geq 0 \). Further, we consider the Lagrange dual problem of minimizing for \( \lambda \geq 0 \) \( \inf_{\xi \in \mathbb{R}^m} g_k(\xi) + \lambda^T (A \xi - b) \) where \( \lambda \in \mathbb{R}^{n_1} \). Since \( g_k(\xi) \) is convex in \( \xi \), together with the assumption that there exists an interior point for the primal problem, then constraint (2.6) is equivalent to the following one:

\begin{equation}
\maximize_{\lambda \geq 0} \ \inf_{\xi \in \mathbb{R}^m} g_k(\xi) + \lambda^T (A \xi - b) \geq 0, \forall k \in \{1, 2, \ldots, K\} .
\end{equation}
Further, constraint (2.7) is equivalent to
\[ \exists \lambda \geq 0, s + \xi^T q + \xi^T Q \xi - y_k^0(x) - y_k(x)^T \xi + \lambda^T (A \xi - b) \geq 0, \forall \xi \in \mathbb{R}^m, \forall k \]
\[ \Leftrightarrow \exists \lambda \geq 0, \quad \left[ \begin{array}{c} 1 \\ \xi^T \end{array} \right] Z_k \left[ \begin{array}{c} 1 \\ \xi \end{array} \right] \geq 0, \forall \xi \in \mathbb{R}^m, \forall k \] \tag{2.8}
\[ \Leftrightarrow \exists \lambda \geq 0, \quad Z_k \geq 0, \forall k \] \tag{2.9}
where the first equivalence is direct from the definition of $Z_k$ and we next prove the latter equivalence. First, it is easy to prove $\Leftarrow$ because of the definition of positive semi-definiteness of a matrix. Then we prove $\Rightarrow$. For any $[\eta_0; \eta] \in \mathbb{R}^{m+1}$ where $\eta_0 \in \mathbb{R}$ and $\eta \in \mathbb{R}^m$. There are two cases: $\eta_0 = 0$ and $\eta_0 \neq 0$. When $\eta_0 = 0$, then $[ \eta_0 \quad \eta^T ] Z_k \left[ \begin{array}{c} \eta_0 \\ \eta \end{array} \right] = \eta^T Q \eta \geq 0$ where the inequality results from the positive semi-definiteness of $Q$. When $\eta_0 \neq 0$, then we have $[ \eta_0 \quad \eta^T ] Z_k \left[ \begin{array}{c} \eta_0 \\ \eta \end{array} \right] = \eta^2 \left[ \begin{array}{c} 1 \\ \eta^T \eta_0 \end{array} \right] Z_k \left[ \begin{array}{c} 1 \\ \eta^T \eta_0 \end{array} \right] \geq 0$ where the inequality is due to (2.8). Therefore, the conclusion of equivalence follows.

To distinguish (2.5) from subsequent approximation schemes, we will refer to it as the original reformulation, hereafter. It is easy to see that problem (2.5) is an SDP problem, which is theoretically solvable in polynomial time. In practice, however, computational challenges remain when the problem size is large. In order to reduce the size of the moment inequality constraint, we employ Principal Component Analysis (PCA) and retain only the most important principal components (PCs). In what follows, we first present the idea of PCA approach from an optimization perspective.

2.2. Low-rank approximation for matrices. At the heart of our proposed methods is exploiting the lower dimensional structure inherent in matrices in practical applications. From a practical perspective, our goal is to approximate a large matrix by a lower dimensional matrix, which will lead to reducing the sizes of the problems. The literature on approximating a matrix with lower dimensional matrix dates back to the seminal paper by Eckart and Young [9].

Eckart and Young showed that given an $m \times n$ matrix $A$, the problem
\[ \min \limits_{\hat{A}} \| A - \hat{A} \|_F \quad \text{subject to} \quad \text{rank}(\hat{A}) \leq r \leq n \]
can be solved using singular value decomposition of the matrix $A$. Let $A = U \Lambda V^T$ be a singular value decomposition of $A$, such that $\Lambda$ is a diagonal matrix with entries $\lambda_i$ and $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$. Let $u_i$ and $v_i$ correspond to the columns of $U$ and $V$ respectively. Define $A_r = \sum_{i=1}^{r} u_i \lambda_i v_i^T$. $A_r$ will be an optimal approximation for $A$ for the Frobenius and spectral norms. That is $\| A - \hat{A} \|_F$ and $\| A - \hat{A} \|_2$.

In the remainder of this paper, we will use singular value decompositions (or specifically for our case eigendecompositions) to approximate matrices in lower dimensions. As stated earlier, low-rank approximations are an active area of research and many other approximations have been proposed based on alternative objective functions and constraints. Our proposed approach is generalizable to these alternative low-rank approximation techniques. Our goal in this paper is to demonstrate how low-rank approximations can be adopted in distributionally robust optimization, and thus in this paper we will restrict our discussions to the most commonly used technique for low-rank matrix approximations.
2.3. PCA approximation for DRO. The eigendecomposition of the positive semidefinite matrix $\Sigma$ can be expressed as follows:

$$\Sigma = U\Lambda U^T = U\Lambda^{\frac{1}{2}}(U\Lambda^{\frac{1}{2}})^T$$

where $U \in \mathbb{R}^{m \times m}$, $\Lambda \in \mathbb{R}^{m \times m}$ is a diagonal matrix and $\Lambda^{\frac{1}{2}}$ replaces diagonal entries of $\Lambda$ with their square roots. Without loss of generality, it is assumed that the diagonal elements of $\Lambda$ are arranged in decreasing order. Then, we introduce another random vector $\xi_I \in \mathbb{R}^m$ whose ambiguity set is

$$D_I(S_I, \mu_I, \Sigma_I) = \left\{ F_I \mid \begin{array}{l} P(\xi \in S_I) = 1 \\ E_{F_I}[\xi] = 0 \\ E_{F_I}[(\xi)(\xi)^T] \preceq I_m \end{array} \right\}$$

where $S_I := \{ \xi_I \in \mathbb{R}^m : U\Lambda^{\frac{1}{2}}\xi_I + \mu \in S \}$ and $I_m$ is the identity matrix of size $m$.

The original problem (2.2) has the same optimal solution as

$$R \in D : \min_{x \in X} \max_{F_I \in D_I} E_{F_I} f(x, U\Lambda^{\frac{1}{2}}\xi_I + \mu)$$

because $\xi := U\Lambda^{\frac{1}{2}}\xi_I + \mu$ has the same support, first and second moments as $\xi$. Further, relying on the ideas behind PCA, we use the leading $m_1$ ($m_1 \leq m$) random variables of $\xi_I$, capturing the dominant variability of $U\Lambda^{\frac{1}{2}}\xi_I$ in $\xi_I$, to approximate $\xi$. Specifically, $\xi \approx U\Lambda^{\frac{1}{2}}[\xi_r, 0_{m-m_1}] + \mu = U_{m \times m_1}\Lambda^{\frac{1}{2}}_{m_1}\xi_r + \mu$ where $\xi_r \in \mathbb{R}^{m_1}$ is the $m_1$-dimensional subvector of $\xi_I$, $0_{m-m_1} \in \mathbb{R}^{m-m_1}$ is a vector whose elements are zero, $U_{m \times m_1} \in \mathbb{R}^{m \times m_1}$ is the $m \times m_1$ upper-left submatrix of $U$, and $\Lambda^{\frac{1}{2}}_{m_1}$ is the $m_1 \times m_1$ upper-left submatrix of $\Lambda^{\frac{1}{2}}$. Then we have the following approximation to (2.10):

$$\min_{x \in X} \max_{F_r \in D_r} E_{F_r} f(x, U_{m \times m_1}\Lambda^{\frac{1}{2}}_{m_1}\xi_r + \mu),$$

where

$$D_r(S_r, \mu_r, \Sigma_r) = \left\{ F \mid \begin{array}{l} P(\xi_r \in S_r) = 1 \\ E_{F_r}[\xi_r] = 0 \\ E_{F_r}[(\xi_r)(\xi_r)^T] \preceq I_{m_1} \end{array} \right\},$$

and $S_r := \{ \xi_r \in \mathbb{R}^{m_1} : U_{m \times m_1}\Lambda^{\frac{1}{2}}_{m_1}\xi_r + \mu \in S \}$.

**Theorem 2.3.** If $f(x, U_{m \times m_1}, \Lambda^{\frac{1}{2}}_{m_1}\xi_r + \mu)$ is $F_r$-integrable for any $F_r \in D_r$, then problem (2.11) has the same optimal value as the following problem:

$$f^*(m_1) := \min_{s, q_r} \quad s + I_{m_1} \bullet Q_r$$

subject to

$$s + \xi_r^T q + \xi_r^T Q_r \xi_r \geq f(x, U_{m \times m_1}, \Lambda^{\frac{1}{2}}_{m_1}\xi_r + \mu), \quad \forall \xi_r \in S_r$$

$$Q_r \succeq 0, \quad x \in X.$$
Proof. The proof of the first part of the results is the same as Theorem 2.1. Let \( \zeta = U_{m \times m} \Lambda_{m_1}^{\frac{1}{2}} \xi_r + \mu \) and we denote its support and ambiguity set by \( \mathcal{S}_\zeta \) and \( \mathcal{D}_\zeta \) respectively. As \( \mathcal{S}_r = \{ \xi_r \in \mathbb{R}^{m_1} : U_{m \times m} \Lambda_{m_1}^{\frac{1}{2}} \xi_r + \mu \in \mathcal{S} \} \) and \( \mathcal{S}_\zeta = \{ \zeta \in \mathbb{R}^m : \zeta = U_{m \times m} \Lambda_{m_1}^{\frac{1}{2}} \xi_r + \mu, \xi_r \in \mathcal{S}_r \} \), it is direct to show that \( \mathcal{S}_\zeta \subset \mathcal{S} \). Moreover, it is straightforward to have \( E[\zeta] = \mu \) and
\[
E[(\zeta - \mu)(\zeta - \mu)^T] = U_{m \times m} \Lambda_{m_1} U_{m \times m}^T
= U \begin{bmatrix}
\Lambda_{m_1} & 0_{m_1,m-m_1} \\
0_{m-m_1,m_1} & 0_{m-m_1,m-m_1}
\end{bmatrix} U^T \preceq U \Lambda U^T = \Sigma,
\]
where \( 0_{a,b} \) is a zero matrix (all of whose elements are zero) of size \( n \times m \). Thus the ambiguity set of \( \zeta \) lies in \( \mathcal{D} \) (the ambiguity set of \( \xi \)). Thus, we have \( \mathcal{D}_\zeta \subset \mathcal{D} \) and further
\[
\max_{F \in \mathcal{D}_\zeta} \mathbb{E}_F f(x, \zeta) \leq \max_{F \in \mathcal{D}} \mathbb{E}_F f(x, \xi).
\]
Therefore the optimal value of problem (2.11) is a lower bound for that of problem (2.2).

Secondly, when \( m_2 > m_1 \), let \( \zeta_1 = U_{m \times m} \Lambda_{m_1}^{\frac{1}{2}} \xi_r + \mu \) and \( \zeta_2 = U_{m \times m_2} \Lambda_{m_2}^{\frac{1}{2}} \xi_r + \mu \) where \( \xi_r \in \mathbb{R}^{m_1}, \xi_r \in \mathbb{R}^{m_2} \). We denote the ambiguity sets of \( \zeta_1 \) and \( \zeta_2 \) by \( \mathcal{D}_1 \) and \( \mathcal{D}_2 \) respectively. Note that
\[
E[(\zeta_2 - \mu)(\zeta_2 - \mu)^T] - E[(\zeta_1 - \mu)(\zeta_1 - \mu)^T] = \sum_{i=m_2+1}^{m_2} U_i U_i^T \geq 0,
\]
where \( U_i \) is the \( i \)-th column vector of matrix \( U \). Additionally, \( \zeta_1 \) has the same mean and support thus \( \mathcal{D}_1 \subset \mathcal{D}_2 \). Therefore
\[
\max_{F \in \mathcal{D}_1} \mathbb{E}_F f(x, \zeta_1) \leq \max_{F \in \mathcal{D}_2} \mathbb{E}_F f(x, \zeta_2).
\]

Finally, we have to show that when \( m_1 \geq \text{rank}(\Sigma) \),
\[
\max_{F \in \mathcal{D}_r} \mathbb{E}_F f(x, U_{m \times m_1} \Lambda_{m_1}^{\frac{1}{2}} \xi_r + \mu) \geq \max_{F \in \mathcal{D}} \mathbb{E}_F f(x, \xi).
\]
For any \( \xi \sim F \in \mathcal{D} \), we assume its covariance matrix is \( \Sigma \preceq \Sigma \), which has the following eigendecomposition:
\[
\tilde{\Sigma} = \tilde{U} \tilde{\Lambda} \tilde{U}^T
\]
We denote the rank of \( \tilde{\Sigma} \) by \( m_2 \) and assume, without loss of generality, that
\[
\tilde{\Lambda} = \begin{bmatrix}
\Lambda_{m_2} & 0_{m_2,m-m_2} \\
0_{m-m_2,m_2} & 0_{m-m_2,m-m_2}
\end{bmatrix}.
\]
Let \( \tilde{\xi} = \tilde{U}^{-1}(\xi - \mu) \) and \( \mathbf{E}[\tilde{\xi}] = 0 \). Then it is easy to show that the covariance matrix of \( \tilde{\xi} \) is \( \tilde{\Lambda} \). Further setting \( \xi := [\xi_{m-m_2}; \xi_{m_2}] \), because of equation (2.15), then the covariance matrix of \( \xi_{m-m_2} \) is a zero matrix and thus \( \xi_{m-m_2} \) is a degenerate random vector and therefore \( \mathbf{P}[\xi_{m-m_2} = 0_{m_2}] = 1 \). Moreover, we have
\[
\mathbb{E}_F f(x, \xi) = \mathbb{E}_{F_{\tilde{\xi}}} f(x, \tilde{U} \tilde{\xi} + \mu) = \mathbb{E}_{F_{\xi_{m_2}}} f(x, \tilde{U}_{m \times m_2} \xi_{m_2} + \mu)
\]
where the second equality exists because of the degeneracy of $\tilde{\xi}_{m-m_2}$. In addition, let 
$\xi_r = \Lambda_{m_2}^{-1} \tilde{\xi}_{m_2}$. It is easy to check that distribution of $\xi_r$ belongs to the distribution of $D_r$. Therefore, equation (2.14) holds.

As Corollary 2.2, for some cases of $f(x, \xi)$ and the support, we have a simpler deterministic version.

**Corollary 2.4.** When the support of $F$ is polyhedral with at least one interior point, i.e., $S = \{x|A^T x \leq b\}$ with $A \in \mathbb{R}^{m_1 \times m}$ and $b \in \mathbb{R}^{m_1}$, if $f(x, \xi)$ is a piecewise linear convex function, precisely, $f(x, \xi) = \max_{k=1}^{K} y_k(x) + y_k(x)^T \xi$ where $y_k(x) = [y_k^1(x), \ldots, y_k^m(x)]$ as well as $y_k(x)$ are affine in $x$ for $k = 1, \ldots, K$, problem (2.12) is reduced to the following problem

$$f^*(m_1) = \min_{x, s, q, \lambda} \{Q_r \}
\begin{cases}
s - y_k^0(x) - \lambda^T b - y_k(x)^T \mu + \lambda^T A \mu \\
q_k + (\Lambda_{m \times m_1} \Lambda_{m_2}^{-1} \Lambda_{m_2}^{-1})^T (A^T \lambda - y_k(x))^T
\end{cases}
\geq 0$$ (2.17a)

$$\forall k \in \{1, 2, \ldots, K\}$$ (2.17b)

$$Q_r \succeq 0, \lambda \geq 0, x \in X.$$ (2.17c)

where $\lambda \in \mathbb{R}^{n_1}$.

**Proof.** The idea of the proof is the same as the proof of Corollary 2.2.

**2.4. Exact reformulation of PCA.** Theorem 2.3 shows that, when the number of principal components is not less than the rank of the upper-bound covariance, i.e., $m_1 \geq \text{rank}(\Sigma)$, the corresponding PCA approximation provides an exact reformulation of problem (2.2). Compared to the original reformulation (2.4), it is not hard to find the exact PCA reformulation has the following computational advantages:

1. When the rank of upper-bound covariances (i.e., $\Sigma$), is full, we consider the PCA approximation with all principal components, i.e., $m_1 = m$, and the corresponding PCA reformulation (2.12), has the same number of variables and constraints as the original reformulation (2.4) respectively. However, the objective function of the PCA reformulation (2.12a) is much sparser than that of the original formulation (2.4a) as there are $m + 1$ variables involved in the objective function of the PCA reformulation whereas there are $m^2 + m + 1$ variables for the original reformulation. The benefit of this sparsity will be numerically validated by the latter numerical experiments.

2. When the rank of upper-bound covariances (i.e., $\Sigma$) is not full, with $m_1 = \text{rank}(\Sigma) < m$, the corresponding PCA reformulation (2.12) not only has a sparser objective but also has fewer variables and constraints, compared with the original reformulation (2.4). For instance, when the support of $F$ is polyhedral with at least one interior point and $f(x, \xi)$ is a piecewise linear convex function in $x$, the corresponding PCA reformulation (2.17) has $1 + n + 2m_1 + m_1^2$ decision variables and the size of its linear matrix inequality is $m_1 + 1$ whereas there are $1 + n + 2m + m^2$ and $m + 1$ respectively for the original reformulation (2.5).

Note that with a sufficient number of principal components, the PCA approximation becomes exact and provides an exact solution with faster runtimes compared with the original formulation. However, if the reduced runtime is not enough for decision makers, the PCA approximation with a smaller number of principal components can further reduce the runtime but at the cost of the quality of solutions. In the next
section, we build the relationship between the number of principal components and the quality of solutions.

2.5. Quality of PCA Solutions. By Theorem 2.3, the PCA approximation yields a relaxation to the original problem. In other words, the optimal objective value of the PCA approximation is less than or equal to that of the original problem. However, Theorem 2.3 does not quantify the gap between them. Here, we present bounds on the gap in the following proposition. The importance of our result is that choosing the number of principal vectors for a specified error bound.

Proposition 2.1. When the support of $F$ is polyhedral with at least one interior point, i.e., $S = \{\xi | A\xi \leq b\}$ with $A \in \mathbb{R}^{n_1 \times m}$ and $b \in \mathbb{R}^{n_1}$, if $f(x, \xi)$ is a piecewise linear convex function, precisely, $f(x, \xi) = \max_{k=1}^{K} y_k^0(x) + y_k(x)^T \xi$ where $y_k(x) = [y_k^1(x), \ldots, y_k^m(x)]$ as well as $y_k^0(x)$ are affine in $x$ for $k = 1, \ldots, K$, then

$$0 \leq f^*(m) - f^*(m_1) \leq \sum_{k=1}^{K} \sum_{i=m_1+1}^{m} \lambda_{i,i} [(A^T \lambda^* - y_k(x^*))^T U_i]^2,$$

where $x^*$ and $\lambda^*$ are optimal solutions of the PCA approximation (2.17) and $f^*(\cdot)$ is defined as in (2.12).

Proof. First, with the results of Theorem 2.3, it is straightforward to show that $f^*(m) - f^*(m_1) \geq 0$. Moreover, according to Corollary 2.4, we have

$$f^*(m) = \min_{x^*, q^r, \lambda^*} x^* + I_{m_1} \cdot Q_r,$$

subject to

$$\begin{bmatrix}
    s - y_k^0(x) - \lambda^T b - y_k(x)^T \mu + \lambda^T A \mu \\
    q_r + (U_{m \times m_1} \Lambda_{m_1}^{1/2})^T (A^T \lambda - y_k(x))^T
\end{bmatrix} \succeq 0
$$

$$\forall k \in \{1, 2, \ldots, K\}$$

$$Q_r \succeq 0, \lambda \succeq 0, x \in X,$$

(2.18)

while

$$f^*(m) = \min_{x^*, q^r, \lambda^*} x^* + I_m \cdot Q_r,$$

subject to

$$\begin{bmatrix}
    s - y_k^0(x) - \lambda^T b - y_k(x)^T \mu + \lambda^T A \mu \\
    q_r + (U \Lambda_2^{1/2})^T (A^T \lambda - y_k(x))^T
\end{bmatrix} \succeq 0
$$

$$\forall k \in \{1, 2, \ldots, K\}$$

$$Q_r \succeq 0, \lambda \succeq 0, x \in X,$$

(2.19)

Let $(x^*, s^*, q^r, \lambda^*, Q_r^*)$ be the optimal solution of problem (2.18). For the sake of clarity, we introduce an auxiliary variable as follows:

$$q_{m-m_1}^k := [U_{m \times m-m_1} (\Lambda_{m-m_1}^{1/2})^T (A^T \lambda^* - y_k(x^*))],$$
where \( \bar{\Lambda}^{m-m_1} \) is the \( m-m_1 \times m-m_1 \) lower-right submatrix of \( \Lambda \). Then we set
\[
q = [q^*_s; 0_{m-m_1,1}],
\]
\[
s = s^* + \sum_{k=1}^{K} \frac{\sqrt{(q^k_{m-m_1})^T q^k_{m-m_1}}}{2} = s^* + \sum_{k=1}^{K} \frac{\sqrt{\sum_{m=m_1+1}^{m} \Lambda_{i,i}[(A^T \lambda^* - y_k(x^*))^T U_i]^2}}{2},
\]
\[
Q = \left[ \begin{array}{ccc}
\sqrt{q^k_{m-m_1} q^{k}_{m-m_1}} & 0_{m_1,1} & 0_{m_1,m_1} \\
0_{m_1,1} & 0_{m_1,m_1} & \sqrt{q^k_{m-m_1} q^{k}_{m-m_1}} \\
\frac{q^k_{m-m_1}}{2} & \frac{q^k_{m-m_1}}{2} & 0_{m-m_1,m_1} \end{array} \right].
\]

Since
\[
\left[ \begin{array}{ccc}
\sqrt{q^k_{m-m_1} q^{k}_{m-m_1}} & 0_{m_1,1} & 0_{m_1,m_1} \\
0_{m_1,1} & 0_{m_1,m_1} & \sqrt{q^k_{m-m_1} q^{k}_{m-m_1}} \\
\frac{q^k_{m-m_1}}{2} & \frac{q^k_{m-m_1}}{2} & 0_{m-m_1,m_1} \end{array} \right] \succeq 0,
\]
It is easy to check that \( (x^*, s, q, \lambda^*, Q) \) is a feasible solution of problem (2.19). Thus
\[
f^*(m) \leq s + I_{m} \bullet Q = f^*(m_1) + \sum_{k=1}^{K} \frac{\sqrt{(q^k_{m-m_1})^T q^k_{m-m_1}}}{2} + \text{trace}(\sum_{k=1}^{K} \frac{q^k_{m-m_1} (q^k_{m-m_1})^T}{2 \sqrt{q^k_{m-m_1} q^k_{m-m_1}}})
\]
\[
f^*(m_1) + \sum_{k=1}^{K} \sqrt{\sum_{i=m_1+1}^{m} \Lambda_{i,i}[(A^T \lambda^* - y_k(x^*))^T U_i]^2}.
\]

Therefore, we have
\[
f^*(m) - f^*(m_1) \leq \sum_{k=1}^{K} \sqrt{\sum_{i=m_1+1}^{m} \Lambda_{i,i}[(A^T \lambda^* - y_k(x^*))^T U_i]^2}.
\]

In Proposition 2.1, some elements of \( q \) are treated as 0. A tighter bound can be obtained as follows,
\[
0 \leq f^*(m) - f^*(m_1) \leq f^*_\text{gap} \sum_{k=1}^{K} \sqrt{\sum_{i=m_1+1}^{m} \Lambda_{i,i}[(A^T \lambda^* - y_k(x^*))^T U_i]^2}
\]
where \( f^*_\text{gap} \) is the optimal value of the following optimization problem
\[
f^*_\text{gap} = \underset{s, q}{\text{minimize}} \sum_{k=1}^{K} \frac{(q^k_{m-m_1} - q)^T (q^k_{m-m_1} - q)}{4s_k} \quad \text{subject to} \quad s_k \geq 0 \quad \forall k \in \{1, 2, ..., K\}
\]
\[
q \in \mathbb{R}^{m-m_1}.
\]

This problem (2.20) can be formulated as a second-order conic programming problem and is thus solvable in polynomial time.

3. Low-rank approximations for chance constraint optimization. In section 2, it was assumed that the uncertainty lies in the objective function. We now turn our attention to the case where the uncertainty lies in the feasible set \( X \); specifically, the constraint function involves uncertainty as follows:
\[
X = \left\{ x \in \mathbb{R}^n : x \in X_0, \min_{F \in \mathcal{D}} P_F \{ h_l^0(x) + h_l(x)^T \xi < 0 \} \geq 1 - \epsilon_l, \forall l = 1, \ldots, L \right\},
\]
\[
\min_{F \in \mathcal{D}} P_F \{ h_l^0(x) + h_l(x)^T \xi < 0 \} \geq 1 - \epsilon_l, \forall l = 1, \ldots, L \},
\]
where \( X_0 \subseteq \mathbb{R}^n \) is a convex closed set that can be represented by semidefinite constraints, \( \xi \in \mathbb{R}^m \) is a random vector with a distribution \( F \), and \( h_i(x) = [h_i^1(x), \ldots, h_i^m(x)] \) as well as \( h_i^0(x) \) are affine in \( x \) for \( i = 1, \ldots, L \). In addition, \( P \) is a probability measure on \( \mathbb{R}^m \) induced by \( \xi \). Constraint (3.1) is called an individual chance constraint where \( \epsilon_i \), \( i = 1, \ldots, L \), are confidence parameters chosen by the decision maker, typically close to zero (e.g. 0.05, 0.10).

**Theorem 3.1.** When the support of \( F \) is polyhedral with at least one interior point, i.e., \( \mathcal{S} = \{ \xi | A \xi \leq b \} \) with \( A \in \mathbb{R}^{n_1 \times m} \) and \( b \in \mathbb{R}^{n_1} \), set \( X \) is equivalent to the following set

\[
\bar{X} := \{ x \in \mathbb{R}^n : x \in X_0, \exists t_l \geq 0, s_l \in \mathbb{R}, q_l \in \mathbb{R}^m, 0 \preceq Q_l \in \mathbb{R}^{m \times m}, \lambda_l, \bar{\lambda}_l \in \mathbb{R}^n_+ \}
\]

subject to

\[
\begin{bmatrix}
    s_l - h^0_l(x) - t_l - b^T \lambda_l \\
    q_l - h_l(x) + A^T \lambda_l \\
    s_l - h^0_l(x) - t_l - b^T \lambda_l \\
    q_l - h_l(x) + A^T \lambda_l
\end{bmatrix}
\begin{bmatrix}
    \Sigma + \mu^T \mu \\
    \Sigma + \mu^T \mu \\
    \Sigma + \mu^T \mu \\
    \Sigma + \mu^T \mu
\end{bmatrix}
\begin{bmatrix}
    t_l \\
    \lambda_l \\
    t_l \\
    \lambda_l
\end{bmatrix}
\leq 0, l = 1, \ldots, L.
\]

It is easy to see that constraints (3.2b) and (3.2c) are linear matrix inequalities. The PCA approximation can be applied to chance constraints in order to reduce the size of the matrix inequality constraints. As in Section 2.3, we replace \( \xi \) by \( U_{m \times m_1} \Lambda_{m_1}^{1/2} \xi_r + \mu \) in constraint (3.1), where the ambiguity set of \( \xi_c \) is still the \( D_r \). According to our results, the PCA approximation of set \( \bar{X} \) is given as sets:

\[
\bar{X}_r(m_1) := \{ x \in \mathbb{R}^n : x \in X_0, \exists t_l \geq 0, s_l \in \mathbb{R}, q_l \in \mathbb{R}^m, 0 \preceq Q_l \in \mathbb{R}^{m \times m_1}, \lambda_l, \bar{\lambda}_l \in \mathbb{R}^n_+ \}
\]

subject to

\[
\begin{bmatrix}
    s_l - h^0_l(x) - h_l(x)^T \mu - t_l - (b - A \mu)^T \lambda_l \\
    (q_l - (U_{m \times m_1} \Lambda_{m_1}^{1/2})^T h_l(x) + (A U_{m \times m_1} \Lambda_{m_1}^{1/2})^T \lambda_l)^T \\
    s_l - (b - A \mu)^T \lambda_l \\
    (q_l + (A U_{m \times m_1} \Lambda_{m_1}^{1/2})^T \lambda_l)^T
\end{bmatrix}
\begin{bmatrix}
    t_l \\
    \lambda_l \\
    t_l \\
    \lambda_l
\end{bmatrix}
\leq 0, l = 1, \ldots, L.
\]

**Proposition 3.1.** The feasible set \( \bar{X}_r(m_1) \) is a subset of \( \bar{X}_r(m_1) \), i.e., \( \bar{X} \subseteq \bar{X}_r(m_1) \). At the same time, \( \bar{X}_r(m_2) \subset \bar{X}_r(m_1) \) if \( m_1 \leq m_2 \). Furthermore, if \( m_1 \geq \text{rank}(\Sigma) \), \( \bar{X}_r(m_1) = \bar{X} \).

**Proof.** The idea of proof is similar to the proof of Theorem 2.3.

These results show that the proposed PCA approach can be extended to solving distributionally robust chance constrained problems. We now present comprehensive computational results to prove the efficacy of our approach in terms of runtime and solution quality.

**4. Experimental Results.** In this section, we compare the performances of the proposed PCA approximation with that of the original formulation on distributionally robust Conditional Value-at-Risk (CVaR) problems. All algorithms are implemented in MATLAB using the modeling language CVX [13, 14] and the corresponding SDP instances are solved using SeDuMi [28] with their default parameters on a computer equipped with a Quad-core Intel Core i7 @ 2.2 GHz processor and 16 GB RAM.
4.1. Distributionally robust CVaR. We consider a distributionally robust version of CVaR problems. CVaR, as an approximation of Value at Risk (VaR), has been extensively studied and applied due to desirable properties like subadditivity and convexity [24]. Additionally, in chance-constrained programming, the CVaR approximation is the least conservative convex approximation of the chance constraints [21]. For more details of CVaR, we refer the reader to Rockafellar and Uryasev [24] and references therein.

Rockafellar and Uryasev [24] proved that the CVaR

\[
1 - \alpha \leq \text{CVaR}_F[g(x, \xi)] 
\]

of a cost function

\[
g(x, \xi) 
\]

can be formulated as the following optimization problem:

\[
\begin{align*}
\text{minimize} & \quad t + \frac{1}{\alpha} E_F [g(x, \xi) - t]^+ \\
\end{align*}
\]

(4.1)

where \( \alpha \in (0, 1) \) is a risk tolerance level; \( F \) is the probability distribution of \( \xi \) and \( \cdot^+ := \max \{0, \cdot\} \). In the case when the exact distribution \( F \) is not readily available but information about the distribution family \( D \) is available, we can consider the following distributionally robust variant of the CVaR problem.

\[
\begin{align*}
\min_{x \in X} \max_{F \in D} \min_{t \in \mathbb{R}} & \quad t + \frac{1}{\alpha} E_F [g(x, \xi) - t]^+ \\
\end{align*}
\]

(4.2)

In what follows, we assume that \( g(x, \xi) = x^T \xi \) and \( X = \{ x \in \mathbb{R}_+^n | \sum_{i=1}^n x_i = 1 \} \) and that the distribution family \( D \) satisfies Assumption 2.1.1. In this case, problem (4.2) is equivalent to the following mini-max problem.

\[
\begin{align*}
\min_{x \in X} \max_{F \in D} \min_{t \in \mathbb{R}} & \quad t + \frac{1}{\alpha} E_F [x^T \xi - t]^+ \\
\end{align*}
\]

(4.3)

The equivalence of (4.2) and (4.3) follows directly from the application of the minimax theorem ([8]) to maximize \( F \in D \) minimize \( \epsilon \in \mathbb{R} \). It is easy to see that \( t + \frac{1}{\alpha} [x^T \xi - t]^+ := \max \{ t, t + \frac{1}{\alpha} (x^T \xi - t) \} \) satisfies Corollary 2.2. Thus, we can solve the CVaR problem exactly using the results of Corollary 2.2 and/or approximately using the proposed PCA approximation scheme.

Here we include two two formulations for completeness. The original formulation of CVaR problems is as follows:

\[
\begin{align*}
\min_{x, t, q, \lambda, Q} & \quad \lambda + \frac{1}{\alpha} [(\Sigma + \mu \mu^T) \bullet Q + \mu^T q + t] \\
\text{subject to} & \quad \begin{bmatrix} t + \frac{1}{\alpha} (q - x)^T \frac{x}{2} Q \\
\frac{q - x}{2} \end{bmatrix} \succeq 0 \\
& \quad \begin{bmatrix} t \\
\frac{q}{2} \end{bmatrix} \succeq 0 \\
& \quad \sum_{i=1}^n x_i = 1, x_i \geq 0, i = 1, ..., n
\end{align*}
\]

(4.4)
while its PCA formulation with all the principal components is:

\[
\begin{align*}
\hat{f}(m) &= \minimize_{x,f,q} \lambda + \frac{1}{\alpha} (\mathbf{1} \bullet Q + t) \\
\text{subject to } &
\begin{bmatrix}
  t + \lambda - \mu^T x \\
  \frac{(q-U\Lambda^2 x)^T}{2} Q
\end{bmatrix} \succeq 0 \\
&
\begin{bmatrix}
  t \\
  \frac{q^T}{2} Q
\end{bmatrix} \succeq 0 \\
&
\sum_{i=1}^n x_i = 1, x_i \geq 0, i = 1, ..., n.
\end{align*}
\] (4.5)

4.2. Comparing exact reformulations. We first conduct numerical experiments to compare the two exact formulations: the PCA reformulation and the original formulation in order to numerically validate the benefits of the PCA reformulation. Here, we only consider the case where the rank of \( \Sigma \) is full. We consider the case when the rank of \( \Sigma \) is not full in Section 4.3. Computational results are presented for randomly generated instances. We set \( \alpha = 0.05 \) and \( n = \{50, 100, 150, 200\} \). The support of the ambiguity set is \( \mathbb{R}^n \), the elements of mean \( \mu \) are uniformly generated on the interval \([5, 10]\), and the covariance matrix \( \Sigma \) is generated randomly using MATLAB function “gallery(’randcorr’,n)”.

For each \( n \), 10 instances are generated and the average performance is shown in Table 4.1 where column one lists the values of \( n \). Columns from two to five present the number of variables in the objective function, the maximum run time, the minimum run time and the average run time for solving the original reformulation of CVaR ((4.4)), respectively. Columns from six to nine present the corresponding results for the PCA reformulation of CVaR in (4.5). The last two columns list the relative gaps of the optimal objective value and the average run time between the PCA reformulation and the original reformulation, respectively.

<table>
<thead>
<tr>
<th>CVaR n</th>
<th># of vars in the objective</th>
<th>Rel. Error</th>
<th>Run time in seconds</th>
<th>Avg. Rel. Imp.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Orig.</td>
<td>PCA</td>
<td>Orig.</td>
<td>PCA</td>
</tr>
<tr>
<td>50</td>
<td>2551</td>
<td>51</td>
<td>0.00</td>
<td>7.2</td>
</tr>
<tr>
<td>100</td>
<td>10101</td>
<td>101</td>
<td>0.00</td>
<td>516.2</td>
</tr>
<tr>
<td>150</td>
<td>22651</td>
<td>151</td>
<td>0.00</td>
<td>7547.2</td>
</tr>
<tr>
<td>200</td>
<td>400201</td>
<td>201</td>
<td>0.00</td>
<td>40768</td>
</tr>
</tbody>
</table>

First, we observe that as \( n \) increases the number of variables in the objective function and CPU times for both reformulations increase. However, note that the number of variables in the objective function of the original reformulation increases quadratically while that of the PCA reformulation increases only linearly. Secondly, the relative gaps of the optimal objective value is 0.00% for all different \( n \) which is consistent with our numerical findings. Last, the relative gap of the average run time shows that the PCA reformulation is computationally more efficient than the original
reformulation. Moreover, the gap increases from 16.00% to 29.19% as \( n \) increases from 50 to 200. Overall, we can infer that the PCA reformulation outperforms the original formulation in terms of runtime.

Next we present some insights into why the PCA reformulation is computationally more efficient than the original formulation. Comparing the two reformulations ((4.4) and (4.5)), we can observe that the main difference among the two reformulations lies in the objective functions, the PCA reformulation has a sparser objective function. To numerically investigate the impact of the objective sparsity, we ran the PCA problem with a different objective function on the same 10 instances for \( n = 100 \). In problem (4.5), the coefficient matrix of variable \( Q \) is \( I \), which is sparse. In the new formulation, we set the coefficient matrix of variable \( Q \) to be a dense matrix whose elements are one in the diagonal and \( 10^{-2} \) elsewhere. With all off-diagonals being much smaller than the diagonals, the underlying problems remains essentially the same, an date numerical structure of the problem is only minimally affected, but the sparsity structure becomes the sam was the original formulation.

The average statistics over 10 instances are shown in Table 4.2, where column one shows the value of \( n \) and column two shows the run time (in seconds) for solving the original reformulation, (4.4). Columns three and four show the run time for solving the original PCA approximation, (4.5), as well as the difference with the original reformulation. Columns five and six show the corresponding results for the PCA problem with the new objective function whose coefficient is now non-sparse.

**Table 4.2:** Average results of PCA formulation with a different objective function.

<table>
<thead>
<tr>
<th>CVAR Origi</th>
<th>PCA</th>
<th>PCA with non-sparse</th>
</tr>
</thead>
<tbody>
<tr>
<td>n Run time(s)</td>
<td>Run time(s)</td>
<td>Run time Diff(s)</td>
</tr>
<tr>
<td>100 532.0</td>
<td>395.1</td>
<td>136.9</td>
</tr>
</tbody>
</table>

From Table 4.2, we can infer that the PCA reformulation is more computationally efficient than the original formulation mainly due to its sparsity of the coefficients of the objective function. Therefore, we can conclude that the PCA reformulation provides decision-makers with a better reformulation compared to the original reformulation.

**4.3. Effect of the number of principle components.** In this section, we focus on the effect of the number of principal components for the PCA approximation. Computational results are presented for both randomly generated instances and instances based on historical data. For randomly generated instances, we set \( n = 100 \) and \( \alpha = 0.05 \). The support of the ambiguity set is \( \mathbb{R}^n \), the elements of mean \( \mu \) are uniformly generated on the interval \([5, 10]\), and the covariance matrix \( \Sigma \) is generated either randomly using MATLAB function “gallery(‘randcorr’,n)” or “partially” randomly. Numerical tests are conducted for \( \Sigma \) with three different ranks, specifically \( \text{rank}(\Sigma) = \{100, 75, 50\} \), which are correspondingly 100%, 75% and 50% of the size of \( \xi \), respectively. For the PCA approximations, the number of the principal components is \( m_1 \in \{100, 75, 50, 25, 10\} \). In the second part of the computational experiments, all means and covariances are estimated using historical market data (obtained from Yahoo Finance) and the tests are conducted using different values for \( \alpha \) and different numbers of principal components.
4.3.1. Results on randomly generated covariance matrices. To randomly generate $\Sigma$, we apply the MATLAB function “gallery(‘randcorr’,n)”, which generates a full-rank matrix. For non-full rank $\Sigma$, we randomly choose some eigenvalues of the generated matrix and set them to be zero to produce the desired rank $\Sigma$. For each different rank $\Sigma$, 10 instances are generated and solved. The main results are shown in Figures 4.1-4.3 and the average statistics over 10 instances are summarized in Table 4.3. Figures 4.1-4.3 show the run time and the relative gap of the optimal objective value using different numbers of principal components for rank 100, 75, and 50 $\Sigma$s. We define the relative gap of the optimal objective value, between the PCA approximation and the original reformulation as $\text{Gap} := \left| \frac{f^*(m) - f^*(m_1)}{f^*(m)} \right| \times 100\%$, where $f^*(m)$ and $f^*(m_1)$ are the optimal values of the original reformulation and the PCA approximation, respectively.

In Figures 4.1-4.3, we display statistics for the relative gaps (GAP) over 10 randomly generated instances for three different ranks of $\Sigma$ (50, 75, and 100). For each boxplot, the minimum, 9% quartile, median, 91% quartile, and maximum are given in order (bottom to top). We also show the average run time (CPU) over the 10 instances. In Table 4.3, column one shows the rank of matrix $\Sigma$ and column two shows the run time for solving the original reformulation. Each set of three columns following shows computational performance for the PCA approximation with varying number of principal components ($m_1 = 100, 75, 50, 25, \text{ and } 10$). The metrics presented are run time (in seconds) (CPU), relative gap (Gap), and theoretical relative gap (Gap2) derived from Proposition 2.1.

<table>
<thead>
<tr>
<th>Rank</th>
<th>CPU (s)</th>
<th>Gap (%)</th>
<th>Gap2 (%)</th>
<th>CPU (s)</th>
<th>Gap (%)</th>
<th>Gap2 (%)</th>
<th>CPU (s)</th>
<th>Gap (%)</th>
<th>Gap2 (%)</th>
<th>CPU (s)</th>
<th>Gap (%)</th>
<th>Gap2 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>553.6</td>
<td>0.00</td>
<td>0.00</td>
<td>73.8</td>
<td>0.89</td>
<td>4.49</td>
<td>6.8</td>
<td>3.18</td>
<td>10.37</td>
<td>0.4</td>
<td>8.14</td>
<td>17.37</td>
</tr>
<tr>
<td>75</td>
<td>523.1</td>
<td>0.00</td>
<td>0.00</td>
<td>74.1</td>
<td>0.00</td>
<td>0.00</td>
<td>6.8</td>
<td>1.60</td>
<td>5.99</td>
<td>0.4</td>
<td>6.44</td>
<td>15.35</td>
</tr>
<tr>
<td>50</td>
<td>447.8</td>
<td>0.00</td>
<td>0.00</td>
<td>58.2</td>
<td>0.00</td>
<td>0.00</td>
<td>7.1</td>
<td>0.00</td>
<td>0.00</td>
<td>0.4</td>
<td>3.20</td>
<td>9.24</td>
</tr>
</tbody>
</table>

Figures 4.1-4.3 and Table 4.3 show that as number of principal components $m_1$ increases the numerical and theoretical gaps decrease and the run time increases. Not surprisingly, the theoretical gap obtained via Proposition 2.1 is larger than the observed gap. This suggests that the theoretical gap may potentially be improved. Secondly, when the number of principal components $m_1$ is larger than or equal to the rank of $\Sigma$, the PCA approximation yields an exact reformulation of the original problem, which supports the results of Theorem 2.3.

For instance, when the rank of $\Sigma$ is 50, only 50 components are needed to obtain the exact optimal solution. But this was accomplished with far less run time, 297 seconds (PCA approximation with 100 components, full rank) compared to 7 seconds (PCA approximation with 50 components). Additionally, the run time for solving the PCA approximations with different numbers of components is less than the run time needed to solve the original reformulation. Even for the full rank $\Sigma$, it took 415 seconds to solve the PCA approximation with 100 components, whereas it took 554 seconds to solve the original reformulation. In other words, when the number of components is the same as the dimension of the random variables, the PCA approximation not only provides an exact reformulation of the original problem, but it is also computationally more efficient. This is mainly due to the sparsity of the coefficients of the...
Fig. 4.1: Performance of the proposed low-rank approximation technique when rank is 100 with 10, 25, 50, 75, 100 components

Fig. 4.2: Performance of the proposed low-rank approximation technique when rank is 75 with 10, 25, 50, 75, 100 components

Fig. 4.3: Performance of the proposed low-rank approximation technique when rank is 50 with 10, 25, 50, 75, 100 components
objective function (see Table 4.2) in the previous section. Finally, it is clear that the proposed PCA approximation allows for practical trade-offs between solution quality and computation time, higher solution quality comes at a cost of increased run time. Furthermore, the PCA approximation provides decision makers with more options to deal with the distributionally robust optimization. For instance, with $\Sigma = 100$, solving the exact reformulation required 500 seconds; however, a high-quality solution within a 5% optimality gap can be found in approximately one minute.

4.3.2. Specially structured covariance matrices. In this section, we consider full-rank specially structured $\Sigma$. With $\lambda_i$ being the $i$-th largest eigenvalue of $\Sigma$, the eigenvalues of $\Sigma$ are created using $\lambda_i = e^{-0.01(i-1)\gamma^{-1}}$, $i = 1, \ldots, 100$ and three different slopes, $\gamma \in \{1, 5, 15\}$. We show the three generating-eigenvalue functions in Figure 4.4. Similar to the previous experiment, 10 test instances are solved for each problem size. The main results are shown by a boxplot in Figure 4.5 and the average statistics are listed in Table 4.4.

In Figure 4.4, we observe that the eigenvalues decrease exponentially. Moreover,
when $\gamma$ increases, the eigenvalue decreases rapidly. From Figure 4.5 and Table 4.4, we obtained similar conclusions for the randomly generated $\Sigma$s computational experiments. For instance, the Gap and the run time are inversely proportional, as Gap decreases computation time increases. Second, the faster the eigenvalues decrease, the smaller the relative gap between the PCA approximation and the original reformulation is, all else being equal. For instance, when the number of principal components is 50, the gap decreases from 2.29% to 0.01% while parameter $\gamma$ increases from 1 to 15. Furthermore, when $\gamma$ is 15, only 50% of the principal components are needed to obtain a high-quality, near optimal solution efficiently (Gap is 0.01% and run time is 6.7 seconds). This computation time is in sharp contrast to the 435.6 seconds run time of the original reformulation. Therefore, it can be concluded that the PCA approximation requires fewer components to obtain high-quality, near-optimal solution for specially structured $\Sigma$s. Finally, even with all the principal components used, the PCA approximation finds the exact solution faster.

### 4.3.3. Historical data induced covariance matrices.

In this section, we evaluate the proposed PCA approximation on CVaR problems using historical market data in 2014-2015 for 123 assets (obtained from Yahoo Finance). The top assets of 10 industry sectors [16] (see Table 4.5) are considered for the portfolio. The mean and variance of 123 returns are estimated using 2014-2015 historical data. We perform the numerical tests using different percentages of principal components, from 100% down to 10% and several values for the confidence parameter $\alpha$, from 0.01 to 0.1. The corresponding results are reported in Table 4.6. The first column shows the percentages of principal components used. Columns two to four (and subsequent sets of three columns) show the optimal value, the relative gap, and the run time, respectively for $\alpha \in \{0.01, 0.05, 0.10\}$.

From Table 4.6, similar conclusions as randomly generated $\Sigma$s and structured $\Sigma$s computational experiments can be drawn. First, as the number of components increases, the relative gap decreases and run time increases. Second, the run time for solving the PCA approximations is less than the run time for solving the original reformulation. Further, the PCA approximation using all principal components yields the same optimal solution as the original reformulation. In other words, the PCA approximation with all components provides a superior reformulation compared to the original reformulation, with exact optimal solution but smaller computation time. For instance, with $\alpha = 0.01$ it took 1,354 seconds to solve the PCA approximation while it took 1,833 seconds to solve the original reformulation. Further, the PCA approximation with 90% of components provides a high-quality, near-optimal solution with a gap of less than 1.4% in less than half the time it took to solve the original
Table 4.5: Tickers of 123 Assets

<table>
<thead>
<tr>
<th>Energy</th>
<th>CMLP</th>
<th>CELP</th>
<th>NTI</th>
<th>KNOP</th>
<th>USAC</th>
<th>MMLP</th>
<th>GMLP</th>
<th>EXLP</th>
<th>DPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Materials</td>
<td>CLMT</td>
<td>NGL</td>
<td>GLOG</td>
<td>MPS</td>
<td>TLP</td>
<td>WPT</td>
<td>TCP</td>
<td>DKL</td>
<td></td>
</tr>
<tr>
<td>Consumer Staples</td>
<td>CISCO</td>
<td>CHSCP</td>
<td>CHSCN</td>
<td>TIS</td>
<td>GYI</td>
<td>PG</td>
<td>SON</td>
<td>KMB</td>
<td>CLX</td>
</tr>
<tr>
<td>Consumer Discretionary</td>
<td>MO</td>
<td>CRWS</td>
<td>RMCF</td>
<td>NTRI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Financials</td>
<td>RSO</td>
<td>TCC</td>
<td>NYMT</td>
<td>SMT</td>
<td>FSEC</td>
<td>RTS</td>
<td>NLY</td>
<td>CSO</td>
<td>MPA</td>
</tr>
<tr>
<td>Healthcare</td>
<td>PDLI</td>
<td>SNH</td>
<td>SBRAP</td>
<td>SBRG</td>
<td>REN</td>
<td>LTC</td>
<td>OHI</td>
<td>AZN</td>
<td>PETS</td>
</tr>
<tr>
<td>Industrials</td>
<td>TAL</td>
<td>SSW</td>
<td>TGR</td>
<td>FLY</td>
<td>ABR</td>
<td>CVA</td>
<td>CTT</td>
<td>PLOW</td>
<td>AYR</td>
</tr>
<tr>
<td>Utilities</td>
<td>APU</td>
<td>CFL</td>
<td>EGAS</td>
<td>NGG</td>
<td>BPF</td>
<td>ED</td>
<td>IE</td>
<td>PEG</td>
<td>EXC</td>
</tr>
<tr>
<td>Technology</td>
<td>MNDO</td>
<td>WILN</td>
<td>CCUR</td>
<td>AREEP</td>
<td>IRM</td>
<td>DFT</td>
<td>DLR</td>
<td>EVOL</td>
<td>GMRN</td>
</tr>
<tr>
<td>Telecoms</td>
<td>CTL</td>
<td>PHI</td>
<td>TLSY</td>
<td>T</td>
<td>BCE</td>
<td>VOD</td>
<td>VZ</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.6: Numerical results of PCA methods with data induced $\Sigma$.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>Value(%)</th>
<th>Gap(%)</th>
<th>CPU(s)</th>
<th>$\alpha = 0.05$</th>
<th>Value(%)</th>
<th>Gap(%)</th>
<th>CPU(s)</th>
<th>$\alpha = 0.1$</th>
<th>Value(%)</th>
<th>Gap(%)</th>
<th>CPU(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>-3.33</td>
<td>-1833.4</td>
<td>1833.4</td>
<td>-1.45</td>
<td>-1729.5</td>
<td>-1491.1</td>
<td>1354.3</td>
<td>-0.99</td>
<td>-1726.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100%</td>
<td>-3.33</td>
<td>0.00</td>
<td>1354.3</td>
<td>-1.45</td>
<td>-0.00</td>
<td>1449.1</td>
<td>-1.00</td>
<td>-1401.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>95%</td>
<td>-3.30</td>
<td>0.90</td>
<td>1001.9</td>
<td>-1.44</td>
<td>0.09</td>
<td>983.4</td>
<td>-0.98</td>
<td>1.01</td>
<td>989.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>90%</td>
<td>-3.29</td>
<td>1.20</td>
<td>717.8</td>
<td>-1.43</td>
<td>1.38</td>
<td>691.1</td>
<td>-1.98</td>
<td>1.01</td>
<td>685.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>80%</td>
<td>-3.19</td>
<td>4.20</td>
<td>360.4</td>
<td>-1.39</td>
<td>4.14</td>
<td>348.4</td>
<td>-0.95</td>
<td>4.04</td>
<td>331.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>70%</td>
<td>-3.12</td>
<td>6.31</td>
<td>170.8</td>
<td>-1.36</td>
<td>6.22</td>
<td>173.0</td>
<td>-0.93</td>
<td>6.06</td>
<td>177.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50%</td>
<td>-2.68</td>
<td>19.52</td>
<td>22.7</td>
<td>-1.17</td>
<td>19.31</td>
<td>23.0</td>
<td>-0.80</td>
<td>19.19</td>
<td>23.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25%</td>
<td>-1.91</td>
<td>42.64</td>
<td>1.0</td>
<td>-0.83</td>
<td>42.76</td>
<td>1.2</td>
<td>-0.57</td>
<td>42.42</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10%</td>
<td>-1.61</td>
<td>51.65</td>
<td>0.3</td>
<td>-0.70</td>
<td>51.72</td>
<td>0.3</td>
<td>-0.48</td>
<td>51.51</td>
<td>0.3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

reformulation. Similarly, we can conclude that the PCA approximation has good performance even for real-life data: the PCA approximation requires fewer components to obtain high-quality, near-optimal solution; with all the principal components used, it is faster for the PCA approximation to find the exact solution.

5. Conclusion. We proposed a computationally efficient approximation for distributionally robust optimization (DRO) problems with moment-based ambiguity sets. Previous results provide a way to reformulate DROs with moment-based ambiguity sets to equivalent semidefinite programs (SDP), which can be solved in polynomial time. However, when the dimensionality of uncertainty is large, corresponding SDP instances become intractable in practice. Thus, new approximation methods that can trade off between solution quality and computation time are needed. For this purpose, we proposed an efficient approximation method based on principal component analysis (PCA), which reduces the dimensionality of the uncertainty space, with minimum loss of information. We showed that the proposed PCA approximation is a relaxation of the original problem when only a subset of the principal components are used. When all principal components are used our method becomes exact, and moreover, provides faster solution times, compared to the original formulation due to the sparsity of its objective function. We also provided a theoretical bound on the differences between the optimal values of original problem and the proposed PCA approximation, which can be used to decide on the number of principal components to use.

Finally, a comprehensive computational study using a distributionally robust
CVaR problem with different covariance structures was conducted to show the strengths of the proposed PCA approximation. Our empirical study showed that the proposed PCA approximation using all principal components outperforms the original reformulation as it yields the same optimal solution with a smaller computation time. For instance, for a problem with 200 variables, our method was 30% faster than the original and the relative improvements increased with increasing problem sizes. We have also observed reductions in run times as we used fewer principal components with minor deviations in accuracy. For instance, using only 50% of the principal components provides a near-optimal solution with a gap of less than 1%, while reducing the run times by orders of magnitude. Such a powerful tool provides decision makers more flexibility to deal with the distributionally robust optimization problem, allowing for direct control of the trade-offs between solution quality and run time.

Acknowledgments. This work was funded by the Laboratory Directed Research and Development (LDRD) program of the Sandia National Laboratories. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-AC04-94AL85000.

REFERENCES