University of Amsterdam
Amsterdam School of Economics
Bachelor Thesis Econometrics and Data Science

Simulation study of data-agnostic and data-dependent random projection methods in compressed regression

Linh Khanh Nguyen (12652105)

University of Amsterdam

Supervisor:	Mario Rothfelder
Date final version:	26th June 2025

This document is written by Linh Khanh Nguyen who declares to take full responsibility for the contents of this document. I declare that the text and the work presented in this document are original and that no sources other than those mentioned in the text and its references have been used in creating it. I have not used generative AI (such as ChatGPT) to generate or rewrite text. UvA Economics and Business is responsible solely for the supervision of completion of the work and submission, not for the contents.

Contents

List of Figures

List of Tables

Abstract

High dimensional data in regression poses significant challenges such as multicollinearity and computational burden. Compressed regression utilizes random projections to map data onto smaller subspaces that are suitable for regression. Using an approximate factor model, a Monte Carlo simulation was conducted to study data-agnostic (GRP, SRP) and data-dependent (DDRP) methods in various scenarios by varying latent factor (r), covariate number (p) and projection dimension (k). Prediction performance is evaluated by ensemble mean squared prediction error. Results revealed that DDRP outperforms GRP and SRP in predictive accuracy and stability under various compression levels. However, the advantage diminishes as k increases, suggesting that for less severe compression, data-agnostic random projections become competitive relative to data-dependent methods.

1 Introduction

Working with data in high-dimensional spaces may lead to undesirable results because of the curse of dimensionality. In the context of regression, as the number of predictors increases, the possibility of linearly dependent covariates increases substantially and lead to multicollinearity. Because of this reason, many dimension reduction techniques exist. The most standard method for linear dimensionality reduction is principal component analysis (PCA), of which extension and application have been well researched (Jolliffe & Cadima, 2016). Despite being a powerful method, PCA is known to be computationally expensive, and interpretations often requires careful attention (James, 2023). Alternatively, random projection (RP) techniques are less computationally expensive, and they have been widely applied in statistics and machine learning (Breger et al., 2020; Li, Vidyashankar, Diao & Ahmed, 2019). The powerful presupposition to RP is the Johnson-Lindenstrauss (JL) lemma, which posits that embedding a high-dimensional subspace to lower dimensions preserves the pairwise relative distance with high probability (Johnson & Lindenstrauss, 1984).

In application, RP is the main component of compressed regression. Compressed regression is a relatively new framework to analyze high-dimensional data, in which the balance between efficiency and interpretability remains an important consideration. Random projection often leads to simpler models with fewer variables, but the selection of the variables is often arbitrary and might eliminate potentially relevant information. As such, the choice of RP matrix is expanding to more data-dependent matrices as supposed to traditional, data-agnostic RP. To investigate the performance between them, this paper builds on the understanding of different random sampling methods of compression matrices either based on the structure of the data or otherwise. This leads to the following research question: How do data-agnostic and data-driven RP compare in predictive

performance when used on compressed linear regression in a high-dimensional setting?

Answering this question can provide a more robust empirical understanding and application of data-aware RP matrix and inform their applications in high-dimensional settings. The research question is tackled by a simulation study, in which different scenarios are generated in which RP methods are applied and compared. The following sections are: section 2 reviews the current developments in compressed regression, section 3 discusses the methodology and simulation procedure, followed by the main results and discussion in section 4, and lastly the conclusion.

2 Literature review

Random projections are mainly used in image and text data pre-processing since the success of research on applied compressed sensing in signal reconstruction by Donoho (2006) and Candès, Romberg and Tao (2006). The core idea of compressed sensing involves reconstructing high-dimensional image from a sparse vector representation, capturing the most important signals through a specified measurement matrix. They identified the conditions under which the compressed lasso estimates asymptotically identify the same non-zeros as the true model and found that the predictions were as good as the lasso estimates on the original data, notions which they named "sparsistence" and "persistence".

Recent developments in compressed regression includes Guhaniyogi and Dunson (2015)'s application of Bayesian inference and model averaging to obtain different weights for the projections based on the strength of influence the compressed variables have on the dependent variable. Built upon Guhaniyogi and Dunson's work, Koop, Korobilis and Pettenuzzo (2019) developed a method for vector autoregression (VAR) based on the same projection matrix and found that Bayesian VAR outperformed other approaches in certain instances, while allowing both scaling and time-variation in the parameters. More recently, others have integrated some form of variable selection through combining full and partial compression estimates with ridge penalty that acts as a tuning parameter (Homrighausen & McDonald, 2020). Bridging compressed regression and random subspace method, Boot and Nibbering (2019) compared random subset regression and Gaussian RP regression and obtained the upper bound for the asymptotic mean squared forecast error, which showed better performance than their benchmark models. Empirically, compressed regression has found application in economics and operational research. For instance, Taveeapiradeecharoen, Chamnongthai and Aunsri (2019) applied Bayesian VAR to forecast foreign exchange rate, while Tsionas, Zelenyuk and Zhang (2025) modeled production technologies using Bayesian compression.

An emerging area of study for RP methods is data-dependent projection. Unlike traditional methods like Gaussian matrix and the computationally optimized sparse matrix which generate matrices independent of the data, data-dependent projection leverage

information found in the data such as structure or variance to improve pitfalls of traditional compressed regression, namely predictive power and interpretability. A recent contribution was developed by Sturges, Yang, Shafaei and Lan (2025), whose algorithm incorporate the data into the row space of a random matrix with independent and identically distributed entries.

3 Methodology

In this section, the theoretical framework for compressed regression will be outlines, in which different data-agnostic and data-driven RP sampling methods are described. For data-agnostic matrices, general Gaussian matrix with i.i.d entries and Achlioptas-type sparse matrix will be discussed (Achlioptas, 2003). Data-driven sampling is based on DDRP method by Sturges et al. (2025), followed by a Monte Carlo study to compare the performance of compressed regression for different sampling methods based on the static approximate factor model by Alessi, Barigozzi and Capasso (2008) with extension of a true model for comparison.

3.1 Compressed regression

I start by describing the setting and defining relevant notation. Consider a standard linear regression model:

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{n\times p}$ is a high-dimensional data matrix with n observations and p covariates ($n \ll p$), $\mathbf{Y} \in \mathbb{R}^n$ is the response vector, $\beta \in \mathbb{R}^p$ is the vector of coefficients, and error terms $\varepsilon \sim N(0, \sigma^2 I)$ are independent and identically distributed. Within the context of compressed regression, a RP (RP) matrix $\mathbf{\Phi} \in \mathbb{R}^{k\times p}$ maps the rows of $\mathbf{X}$ from $\mathbb{R}^p$ to $\mathbb{R}^k$, effectively reducing the number of covariates from n to p. The resulting model specification is

$$\mathbf{Y} = \tilde{\mathbf{X}}\beta^c + \varepsilon \tag{2}$$

where $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{\Phi}^\top$ and $\beta^c \in \mathbb{R}^k$. Here, $\tilde{\mathbf{X}}$ is the compressed design matrix, and β^c is the vector of coefficients for the compressed model. Compressing $\mathbf{X}$ to $\tilde{\mathbf{X}}$ reduces the number of features, which mitigates overfitting and stabilizes linear regression estimation. Concerning the distortion of information when projecting onto a smaller subspace, the Johnson-Lindenstrauss (JL) lemma guarantees that the distance between pairs of points are preserved. The lemma is discussed in more details in the next sub-section.

3.1.1 Data-agnostic RP

The goal of RP is to trade a controlled amount of accuracy with a smaller model size. Let $0 < \varepsilon < 1$, and let X be a set of n points in $\mathbb{R}^p$. When $k = \mathcal{O}(\log(n)\varepsilon^{-2})$, there exists a linear map $f : \mathbb{R}^p \to \mathbb{R}^k$ such that for all pairs of points $(u, v) \in X$, the squared Euclidean distance between them is approximately preserved under f with high probability (Johnson & Lindenstrauss, 1984):

$$(1-\varepsilon)||u-v||^2 \leq ||f(u)-f(v)||^2 \leq (1+\varepsilon)||u-v||^2 \tag{3}$$

The JL lemma is important for compressed regression, as it ensures the relationship between the observations, thus the relevant structure for regression, is preserved despite drastic dimension reduction. This theoretical guarantee justifies the validity of performing regression in compressed space. The common implementation of JL lemma is Gaussian random matrix. Each element is sampled according to $\Phi_{ij} \sim N(0, 1)$. With the entries defined, the mapping

$$f(\mathbf{X}) = \frac{1}{\sqrt{k}}\mathbf{X}\mathbf{\Phi}^\top \tag{4}$$

preserves norm in (3) with $k = 20\log(n)\varepsilon^{-2}$ and $(u, v) \in X \subset \mathbb{R}^p$ (Casarin & Veggente, 2021).

One may choose the degree of sparsity for entries in RP when computational efficiency is a focus of the research. In Achlioptas (2003), this RP formulation allows sparsity while maintaining JL property:

$$\Phi_{ij} = \sqrt{s} \cdot \begin{cases} +1 & p = \frac{1}{2s} \\ 0 & p = 1 - \frac{1}{s} \\ -1 & p = \frac{1}{2s} \end{cases} \tag{5}$$

The factor $\sqrt{s}$ controls for the variance. A common choice is $s = 3$, where two-thirds of the entries are zero. Sparse RPs offer computational efficiency over Gaussian, but it may introduce distortion in the regression estimates, especially when the target dimension k is very small.

3.1.2 Data-driven RP

In contrast, a data-driven projection by construction incorporates structure of $\mathbf{X}$ into the RP matrix. Following Sturges et al. (2025), DDRP's mechanism starts with generating a matrix $\mathbf{W} \in \mathbb{R}^{k\times n}$ with i.i.d entries from $N(0, 1/k)$, from which the RP is computed as:

$$\mathbf{\Phi} = (\mathbf{W}\mathbf{X})^\top.$$

The researchers found that D^{2}RP distorts the data distance relatively more than $[1-\varepsilon, 1+\varepsilon]$ in (3) of data-independent variants. They suggested that reconstruction of the projected data is possible and the modeling risk decays as k increases.

3.2 Monte Carlo study

To evaluate the performance of compressed regression under high-dimensional settings, I stimulate data using the approximate factor model data-generating process (DGP) from Alessi et al. (2008). This framework provides a widely accepted structure where a small number of latent factors explain most variation in the observed data. The DGP allows us to assess how well different RPs retain essential predictive elements in the presence of noise and dimensionality.

3.2.1 Data Generating Process

To create the data, the design matrix $\mathbf{X}$ is generated independently by sampling according to a static factor model. For $i = 1, ..., n$ and $j = 1, ..., p$, where i is the number of observations and j is the number of covariates, the model simulates the data matrix according to Alessi et al.'s (2008) DGP1:

$$x_{ij} = \sum_{k=1}^{r} \lambda_{ik} F_{jk} + \sqrt{\theta} e_{ij} \tag{6}$$

where

$$F_{jk}, \lambda_{ik} \sim N(0,1)$$

$$e_{ij} \sim N(0,1) \quad \text{and} \quad r = \theta$$

where F_{jk} is the factors, λ_{ij} is the factor loadings, θ is the scaling factor for the error term, and r is the number of latent variables underlies the data structure. The model assumes homoskedasticity, with the common component $\sum_{k=1}^{r} \lambda_{ik} F_{jk}$ and the noise component $\sqrt{\theta} e_{ij}$ having equal variance. The parameter θ controls the variance of the idiosyncratic error term. In this study, we set $r = \theta$ which ensures that as the data complexity increases, the overall noise level in the data also increases proportionally. The result is a $n \times p$ matrix.

In extension to the model, the response vector is generated so that the signals in $\mathbf{X}$ is proportional to the signals in β_*. Then the response vector is generated as $\mathbf{Y} = \mathbf{X}\beta_* + \eta$ where η_i are i.i.d with mean zero and unit variance. The vector of true coefficients β_* is defined as

$$\beta_* = \frac{1}{\sqrt{(r+\theta)p}} \iota$$

where $\iota = (1, ..., 1)^\top \in \mathbb{R}^p$. The choice of β_* is derived from setting the signal-to-noise ratio to be one, irrespective of r and θ. This allows a fair comparison across different

scenarios with varying structure and noise level. The derivation of β_* is presented in the Appendix B.

3.2.2 Simulation Procedure

In order to evaluate the out-of-sample predictive performance of compressed regression when applying different RP methods, the procedure is as followed. For each Monte Carlo trial, $\mathbf{X}$ and $\mathbf{Y}$ are generated randomly. Then, for each RP method, 100 distinct RP matrices $\mathbf{\Phi}$ are generated. These matrices are applied to $\mathbf{X}$ to create $\tilde{\mathbf{X}}$, and ordinary least squares (OLS) is performed on the projected subspace to obtain 100 sets of coefficient estimates. An ensemble prediction or that trial is then computed by averaging the predictions derived from these 100 estimated models. This entire process (DGP, 100 RP matrices, ensemble prediction) is repeated for 100 Monte Carlo simulations to obtain a robust estimate of the mean MSPE and its standard deviation for each method. This simulation procedure is carried out over 18 scenarios.

First, the number of observation is fixed to $n = 100$, and three values of p is chosen: 110, 200, 1000. The choices of p represents increasing level of high-dimensionality from small to very high. For a given latent variable $r \in \{1, 3, 5, 8, 10, 15\}$, the factor matrix F and factor loading matrix Λ are drawn. The latent variable r controls the signal structure in the data, as well as scaling the variance of design matrix $\mathbf{X}$. Lastly, the variable of targeted dimension $k = \{3, 8, 15, 30\}$ were chosen. The choices of n, p, r and k were designed to study cases where the targeted dimension is larger, almost equal to, and smaller than the latent structure.

3.2.3 Evaluation Metrics

For each pair of (n, p, r), I compute the mean squared prediction error (MSPE) and report the average error, standard deviation, and 95% confidence interval across 100 Monte Carlo simulations. Let $\hat{y}_i$ denotes the predicted value and y_i the true response for the i-th observation in a test set of size n. The prediction accuracy is quantified by the MSPE:

$$\text{MSPE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{7}$$

For each method, the MSPE of 100 simulations will be averaged. The 95% confidence interval over the Monte Carlo is calculated with:

$$\text{CI} = \text{mean} \pm 1.96 \times \frac{\text{standard deviation}}{\sqrt{100}} \tag{8}$$

4 Results and discussion

4.1 Dependence on latent variables r

Table 1 displays the MSPE and the standard deviations of compressed regression based on 3 different RP methods (Gaussian RP (GRP), sparse RP (SRP), and data-dependent RP (DDRP)), with sample size $n = 100$, covariates $p = 110$ and projection dimension $k = 8$ across 6 scenarios of the latent variable r.

At the lowest latent dimension $r = 1$, indicating a sparse and weak inherent structure of underlying data, all methods suffer from significant error, resulting in high MSPEs. However, DDRP showed the lowest MSPE of 1.183, which is a 1.7% improvement over both GRP and SRP (1.204). This suggests that despite having minimal inherent structure, incorporating data into a random matrix can still provide a slight advantage in signal extraction.

	r = 1		r = 3		r = 5	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	1.204	0.166	1.469	0.203	1.617	0.240
Sparse	1.204	0.166	1.468	0.204	1.618	0.240
DDRP	1.183	0.161	1.407	0.203	1.507	0.214
	r = 8		r = 10		r = 15	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	1.752	0.279	1.866	0.249	1.980	0.287
Sparse	1.746	0.274	1.870	0.246	1.981	0.293
DDRP	1.607	0.246	1.703	0.218	1.808	0.246

Table 1: MSPE and standard deviation of predicted responses. From simulation with $n = 100$, $p = 110$, $k = 8$.

As the number of latent variables r increases, indicating a more complex underlying data structure, a consistent increase in MSPE is observed across all three methods. This could be attributed to the increasing variance in the DGP equation (6) as r is set to be equal to θ in this case. However, despite increasing noise in the model, DDRP consistently maintained the lowest MSPE and the lowest standard deviation regardless of r. For instance, at $r = 10$, DDRP's MSPE of 1.703 is notably lower than GRP's 1.866 and SRP's 1.870, with standard deviation of 0.218, 0.249 and 0.246, respectively. The results suggest that, at $k = 8$, DDRP yielded more accurate predictions as well as offered greater stability in performance compared to its data-agnostic counterparts as r increases.

4.2 The effect of p

The simulation results for $(n, p) = \{(100, 200), (100, 1000)\}$ at $k = 8$ are showed in Table 2 and Table 3 respectively.

The analysis across increasing p reveals a clear and expected trend: the growth in dimensionality universally worsen prediction errors across all RP methods. For instance, when p increases nearly tenfold (from 110 to 1000), the average MSPE for GRP at $r = 15$ increases by 5.3%, from 1.980 to 2.084. Similarly, the MSPE for DDRP in the same scenario increases by about 6.4%, from 1.808 to 1.923, demonstrating a slightly bigger increase.

Visualizing the tables, Figure 1 uses $p = 110$ as the baseline and compares the

	r = 1		r = 3		r = 5	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	1.260	0.155	1.531	0.199	1.684	0.213
Sparse	1.260	0.154	1.532	0.199	1.681	0.213
DDRP	1.244	0.153	1.467	0.194	1.566	0.218
	r = 8		r = 10		r = 15	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	1.851	0.256	1.909	0.302	2.002	0.286
Sparse	1.855	0.266	1.914	0.299	2.007	0.292
DDRP	1.699	0.258	1.721	0.242	1.829	0.244

Table 2: MSPE and standard deviation of predicted responses. From simulation with $n = 100$, $p = 200$, $k = 8$.

	r = 1		r = 3		r = 5	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	1.248	0.175	1.570	0.210	1.729	0.269
Sparse	1.248	0.171	1.569	0.208	1.732	0.276
DDRP	1.242	0.170	1.519	0.198	1.598	0.251
	r = 8		r = 10		r = 15	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	1.937	0.296	2.006	0.329	2.084	0.280
Sparse	1.943	0.300	2.011	0.333	2.086	0.279
DDRP	1.752	0.268	1.799	0.268	1.923	0.248

Table 3: MSPE and standard deviation of predicted responses. From simulation with $n = 100$, $p = 1000$, $k = 8$.

relative performance of each method when p increases. It can be observed that all methods shows similar trends as we increase the number of covariates, but notably GRP and SRP have very comparable decay in performance. Surprisingly, Figure 1 suggests that DDRP's performance may decay slightly faster when p increases as opposed to GRP and SRP. However, it is important to note that except for $k = 30$, DDRP's MSPEs remain lower.

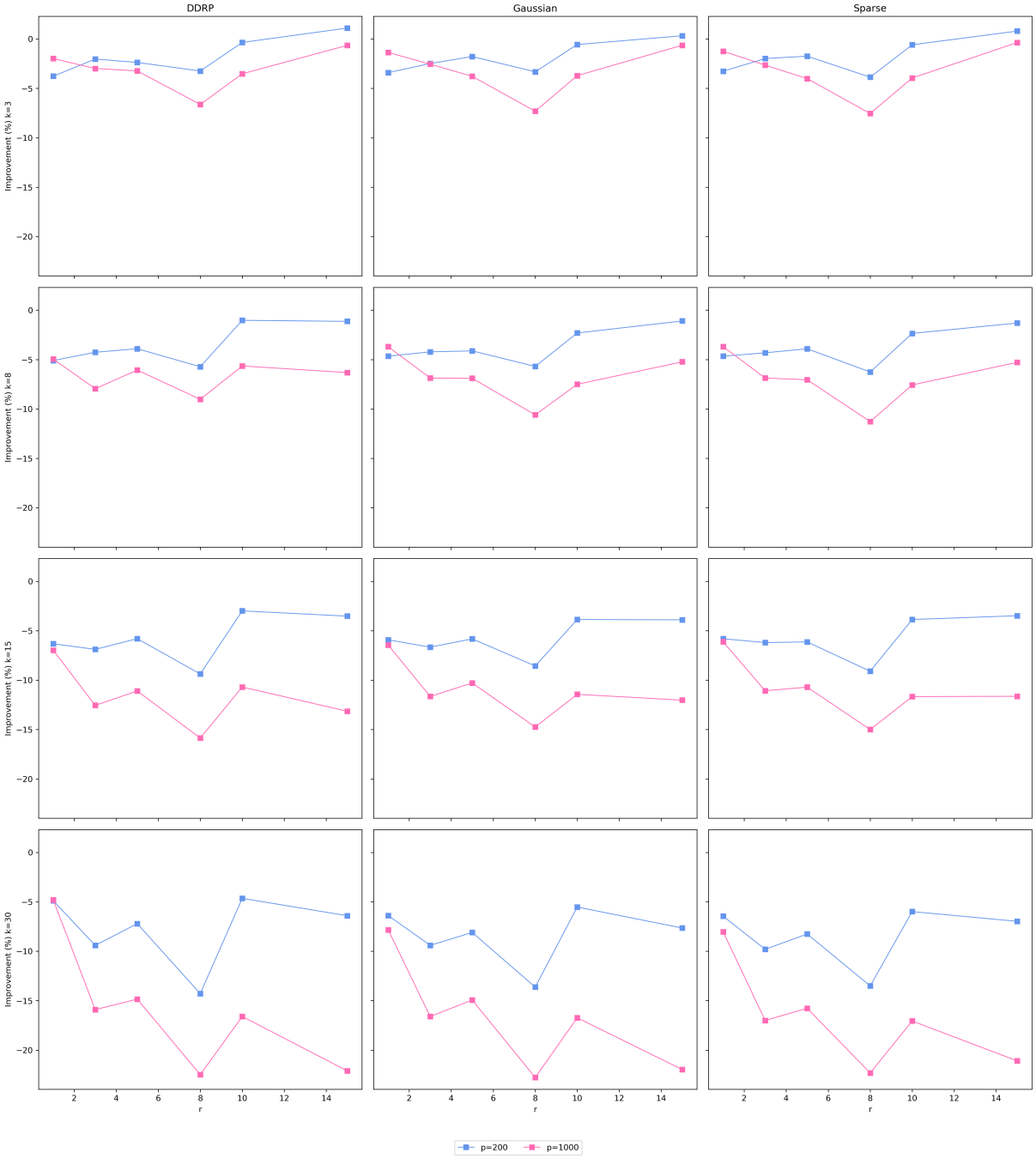

Figure 1: Percentage change in MSPE for all methods when $p = 200$ and $p = 1000$, relative to the baseline scenario of $p = 110$. Results are shown for various latent factor dimensions r and projection dimensions k.

An interesting observation is that across all cases, it can be observed that there is a dip in performance around $r = 8$ and $r = 10$ following by recovery for $k = 3$ and $k = 8$. A potential explanation could be that the underlying structure becomes more "evident" as r increase, it becomes more effectively captured by RP, leading to a relative improvement. This possibly highlights the interplay between inherent data structure and dimensionality, which in turn influence how RP can preserve information. However, the apparent decrease at $r = 8$ may require further inspection.

In general, the phenomenon we observe in this section exemplifies the impact of the curse of dimensionality, where increased feature space hinders signal exploration and risks incorporating irrelevant noise into the model. Despite this challenge, DDRP consistently maintains comparatively lower errors and often lower standard deviations than GRP and SRP, especially in the most extreme $p = 1000$ case. This performance highlights DDRP's superior ability to preserve information after compression, which is a highly desirable characteristic for predictive modeling in extremely high-dimensional cases.

4.3 Sensitivity analysis over varying k

To analyze the properties of the methods at different projection dimensions k, the plot of the mean MSPEs from 100 simulations against r for different choices of k is given in Figure 2. Furthermore, using GRP as the baseline model, Figure 3 visualizes the percentage improvement in MSPE where a positive percentage indicates better performance than GRP while a negative value indicates the opposite. Lastly, the percentage improvement in standard deviations is shown in Figure 4. The exact MSPEs and standard deviations are displayed in Appendix A.

As expected, for the most restrictive compression $k = 3$, all methods exhibits higher MSPE compared to scenarios with larger k. At very low projection dimension, the information loss is substantial. Despite being under restrictive conditions, however, DDRP consistently yields lower MSPEs than both GRP and SRP, particularly when k approaches the inherent r. For instance, at $p = 110$, DDRP achieves a significant improvement of over 10% for $r = 3$ and $r = 5$. This suggests that mapping data onto the RP matrix allows retention of relatively higher proportion of predictive signal even when k is a considerable restriction.

As shown in Figure 3, when k increases to 8, DDRP's improvement remains strong at around 2-10%. Subsequently when $k = 15$, the improvement is slightly diminished, around 2-8%, though this is still a significant improvement in performance. The gaps between DDRP and data-agnostic approaches, as can be observed in Figure 3, becomes pronounced at certain values of r. Notably, when k is approximately equal to r, DDRP demonstrates the most improvement, although this effect diminishes as r exceeds k.

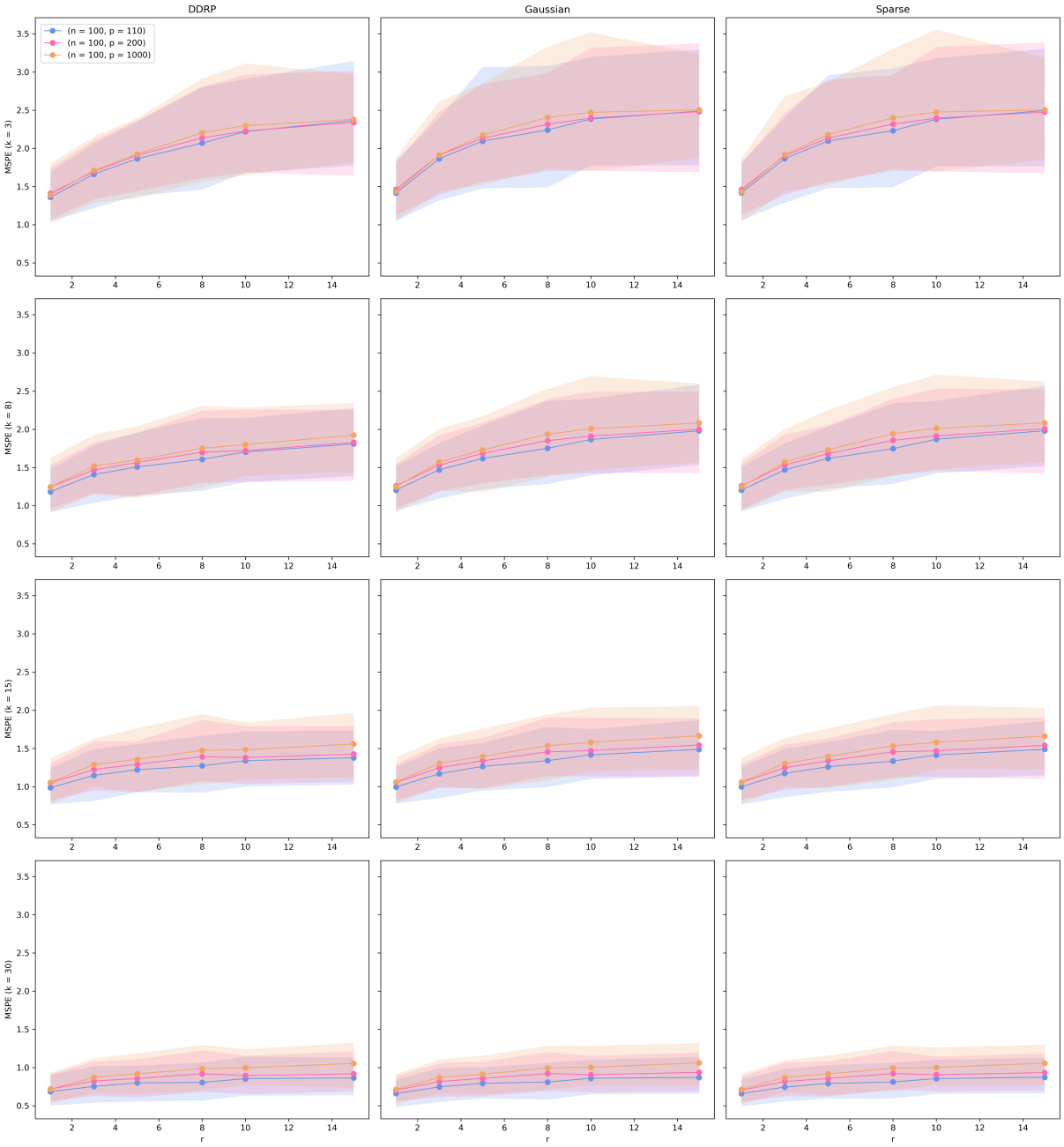

Figure 2: Plots of the mean squared prediction errors and 95% CI of GRP, SRP and DDRP methods. From the simulation with $r = \{1, 3, 5, 8, 10, 15\}$, for varying projection dimensions $k = \{3, 8, 15, 30\}$.

On the other hand, SRP's performance is mixed as seen through the fluctuation between the baseline. It shows negligible and sometimes slightly worse performance, though this is to be expected as the advantage of SRP lies in the computational efficiency that it offers while maintaining comparable loss of information to GRP.

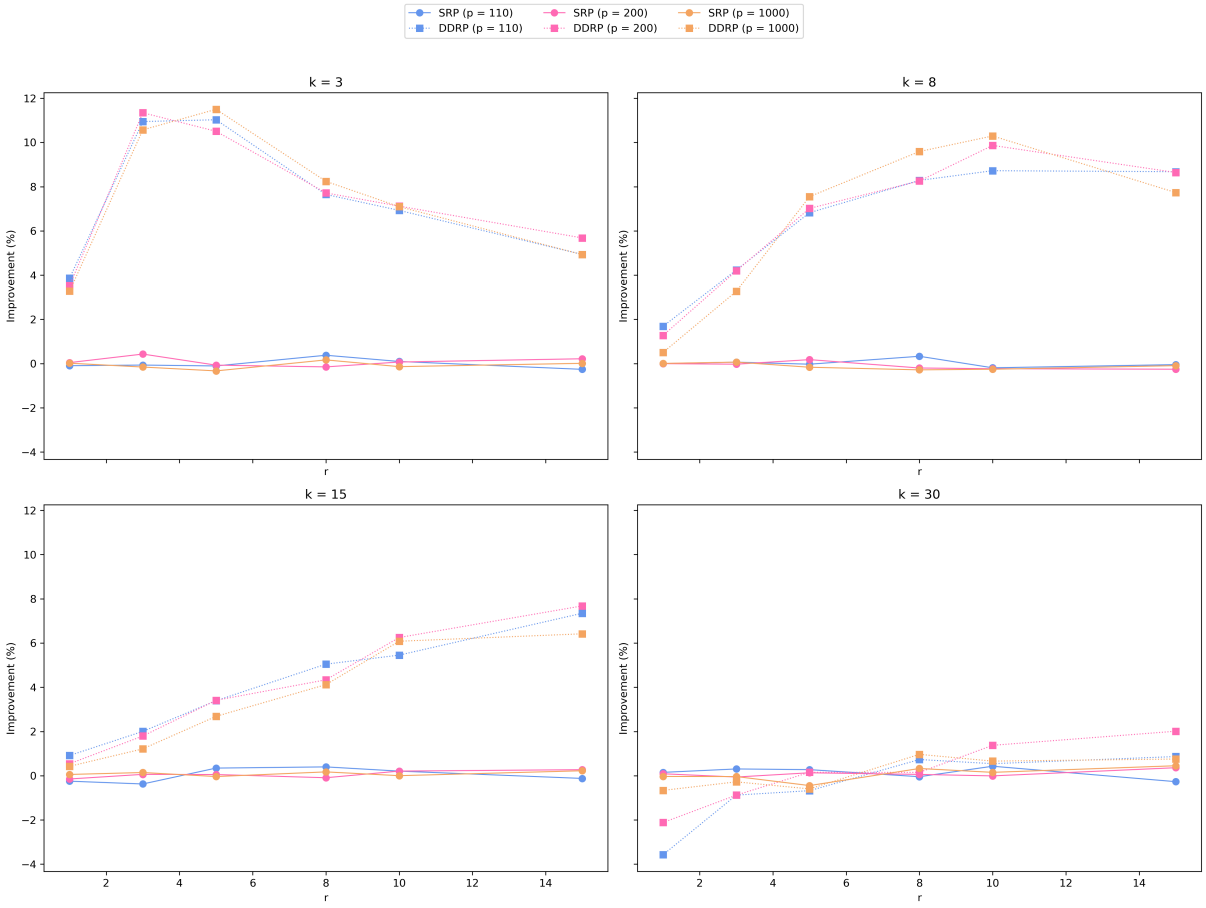

Figure 3: Percentage change in mean squared prediction error of DDRP and SRP relative to GRP. From the simulation with $r = [1, 3, 5, 8, 10, 15]$, $k = [3, 8, 15, 30]$.

At $k = 30$, the advantage we observed in previous cases for DDRP became less clear, occasionally becomes a disadvantage when $r \ll k$. When k is small, DDRP has the ability to prioritize important directions of data structure into the projection matrix and therefore it is able to perform better than purely random choices as in GRP and SRP. When $k = 30$, we are essentially retaining a much larger fraction of the original covariate space compared to when $k = 3$ or $k = 8$. As a result, the performance gap between DDRP and data-agnostic counterparts narrowed considerably. This may occur as for larger k, even purely RP can preserve enough relevant information, making the data-dependent RP less impactful. This result is as expected from JL lemma, which posits that as k increases, the probabilistic guarantee of distance preservation is stronger, making the choice of RP matrix less significant.

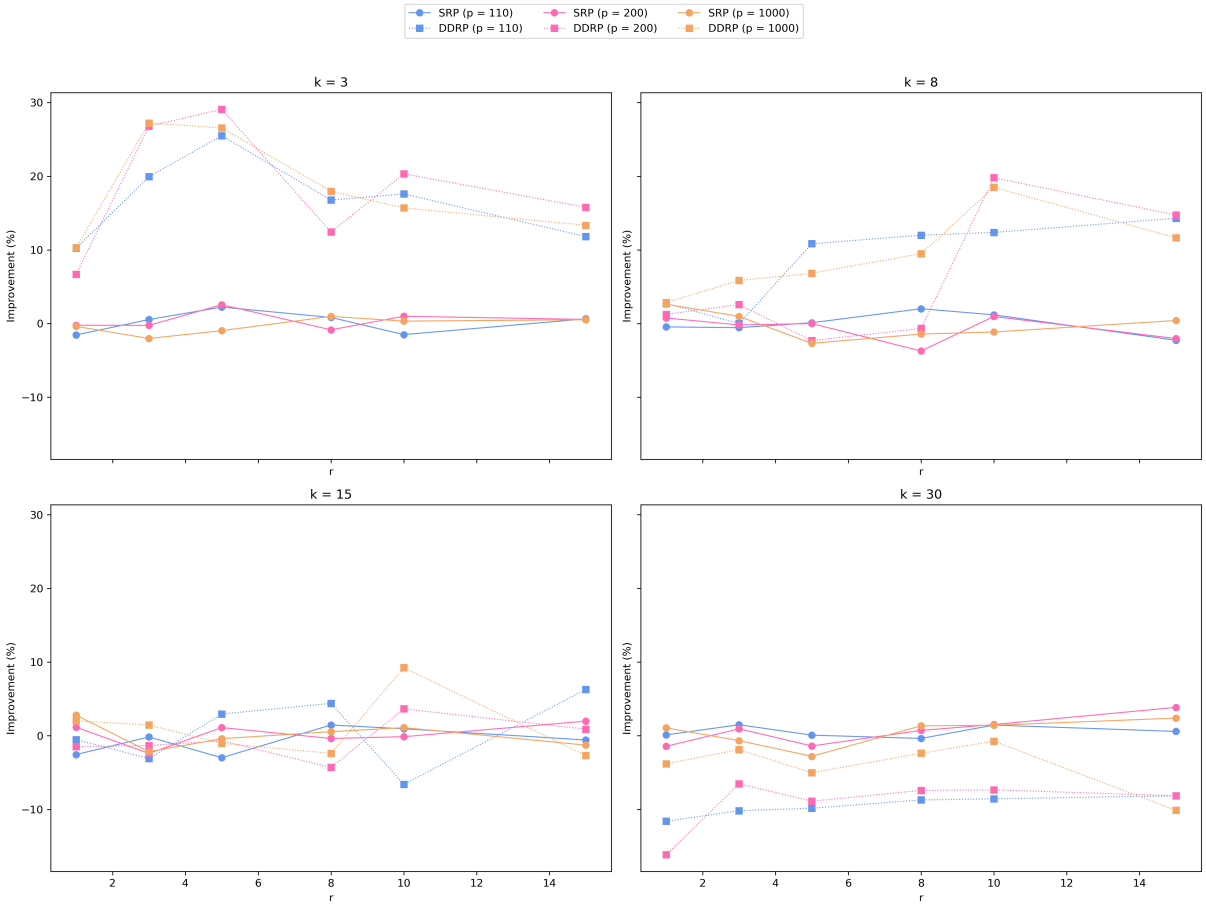

Figure 4: Percentage change in standard deviation of predicted responses for DDRP and SRP relative to GRP. From the simulation with $r = \{1, 3, 5, 8, 10, 15\}$, for varying projection dimensions $k = \{3, 8, 15, 30\}$.

Beyond predictive accuracy, the stability of the model's prediction can be observed from Figure 4. At $k-3$, DDRP's advantage is most prominent, exceeding 20% for certain cases. As k increases, again we could see a diminishing stability when compared to GRP and SRP. This reinforces the idea that when less compression is needed, the need for DDRP lessens.

In summary, the analysis of percentage change in MSPE and standard deviation strengthens out understanding of DDRP's performance. The data-dependent approach enables DDRP to achieve significantly more stable and consistent predictions, particularly under high compression when other methods struggle.

5 Conclusion

This paper leveraged a comprehensive simulation study to empirically evaluate and compare the predictive performance of GRP, SRP and DDRP matrices within the framework of compressed linear regression in high-dimensional settings. I have replicated and extended upon the model proposed by Alessi et al. (2008) that allowed freedom in choosing the number of latent variables that control the inherent structure of the data. In addition to that, a true model with derivation of β that follows r and θ was added to create the true model for comparison.

The result from the simulation reveal very promising capabilities of compressed regression based on data-dependent algorithm. The DDRP seems to perform the best overall across all scenarios explored even when the number of covariates increased exponentially. Data-dependent RP performed the best when the number of inherent structure is about as equal as the number of targeted dimension, when the targeted dimension is small. When k increases however, the traditional data-agnostic RP methods become comparably good choice.

This research contributes to the growing understanding of compressed regression by providing empirical evidence for the advantages of data-dependent RP compared to two of the most common RP methods. Specifically, it offers insights under controlled environments, demonstrating that DDRP offers substantial benefits in certain high-dimensional settings.

Despite these insights, this study is subjected to limitations. The simulation relies on a specific DGP and assumes linearity between predictors and response, whereas real-life data may exhibit nonlinearities or different underlying structure. Furthermore, the study's primary performance indicator was MSPE. More in depth analysis such as computational efficiency across each specification may offer additional insights regarding the accuracy-efficiency trade-off. Moreover, the study limited k to be at maximum of 30 due to the choice of $n = 100$, therefore restricting the analysis.

In future work, there are several possible directions that can be taken. One could explore the performance of more complex models such as non-linear regression and their computational efficiency. Additionally, one could expand on the choices of n, p and k, or apply this research on real high-dimensional datasets for practical considerations.

References

Achlioptas, D. (2003). Database-friendly random projections: Johnson-lindenstrauss with binary coins. *Journal of Computer and System Sciences*, *66*(4), 671–687. doi: 10.1016/S0022-0000(03)00025-4

Alessi, L., Barigozzi, M. & Capasso, M. (2008, May). *A robust criterion for determining the number of static factors in approximate factor models* (Working Paper Series No. 903). European Central Bank.

Boot, T. & Nibbering, D. (2019). Forecasting using random subspace methods. *Journal of Econometrics*, *209*(2), 391–406. doi: 10.1016/j.jeconom.2019.01.009

Breger, A., Orlando, J. I., Harar, P., Dörfler, M., Klimscha, S., Grechenig, C., ... Ehler, M. (2020). On orthogonal projections for dimension reduction and applications in augmented target loss functions for learning problems. *Journal of Mathematical Imaging and Vision*, *62*(3), 376–394. doi: 10.1007/s10851-019-00902-2

Candès, E. J., Romberg, J. K. & Tao, T. (2006). Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, *59*(8), 1207–1223. doi: 10.1002/cpa.20124

Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, *52*(4), 1289–1306.

Guhaniyogi, R. & Dunson, D. B. (2015). Bayesian compressed regression. *Journal of the American Statistical Association*, *110*(512), 1500–1514. doi: 10.1080/01621459.2014.969425

Homrighausen, D. & McDonald, D. J. (2020). Compressed and penalized linear regression. *Journal of Computational and Graphical Statistics*, *29*(2), 309–322. doi: 10.1080/10618600.2019.1660179

James, G. (2023). *An introduction to statistical learning: with applications in python* (1st ed.). Springer International Publishing. doi: 10.1007/978-3-031-38747-0

Johnson, W. B. & Lindenstrauss, J. (1984). Extensions of lipschitz mappings into a hilbert space. *Contemporary Mathematics*, *26*, 189–206.

Jolliffe, I. T. & Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A*, *374*(2065), 20150202. doi: 10.1098/rsta.2015.0202

Koop, G., Korobilis, D. & Pettenuzzo, D. (2019). Bayesian compressed vector autoregressions. *Journal of Econometrics*, *210*(1), 135–154. doi: 10.1016/j.jeconom.2018.11.009

Li, L., Vidyashankar, A. N., Diao, G. & Ahmed, E. (2019). Robust inference after random projections via hellinger distance for location-scale family. *Entropy*, *21*(4), 348. doi: 10.3390/e21040348

Sturges, J., Yang, L., Shafaei, S. & Lan, C. (2025). Efficient data-dependent random

projection for least square regressions. In *Icassp 2025 - 2025 ieee international conference on acoustics, speech and signal processing (icassp)* (p. 1-5). doi: 10.1109/ICASSP49660.2025.10889832

Taveeapiradeecharoen, P., Chamnongthai, K. & Aunsri, N. (2019). Bayesian compressed vector autoregression for financial time-series analysis and forecasting. *IEEE Access*, *7*, 16777–16786. doi: 10.1109/ACCESS.2019.2895022

Tsionas, M., Zelenyuk, V. & Zhang, X. (2025). Goodness-of-fit in production models: A bayesian perspective. *European Journal of Operational Research*. doi: 10.1016/j.ejor.2025.01.030

A Results from simulation

A.1 With k = 3

	r = 1		r = 3		r = 5	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	1.415	0.210	1.865	0.295	2.094	0.396
Sparse	1.416	0.213	1.866	0.294	2.096	0.387
DDRP	1.360	0.188	1.661	0.236	1.863	0.295
	r = 8		r = 10		r = 15	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	2.240	0.421	2.384	0.372	2.488	0.420
Sparse	2.232	0.418	2.381	0.378	2.495	0.417
DDRP	2.069	0.351	2.218	0.307	2.365	0.371

Table A1: MSPE and standard deviation of predicted responses. From simulation with $n = 100$, $p = 110$, $k = 3$.

	r = 1		r = 3		r = 5	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	1.463	0.187	1.911	0.309	2.131	0.343
Sparse	1.462	0.187	1.903	0.310	2.133	0.334
DDRP	1.411	0.174	1.694	0.226	1.907	0.243
	r = 8		r = 10		r = 15	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	2.315	0.366	2.397	0.460	2.480	0.413
Sparse	2.318	0.369	2.395	0.456	2.475	0.411
DDRP	2.136	0.321	2.226	0.367	2.339	0.348

Table A2: MSPE and standard deviation of predicted responses. From simulation with $n = 100$, $p = 200$, $k = 3$.

	r = 1		r = 3		r = 5	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	1.434	0.211	1.912	0.312	2.173	0.406
Sparse	1.433	0.212	1.915	0.318	2.180	0.410
DDRP	1.387	0.190	1.710	0.227	1.923	0.298
	r = 8		r = 10		r = 15	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	2.404	0.423	2.472	0.475	2.504	0.370
Sparse	2.400	0.419	2.475	0.473	2.504	0.368
DDRP	2.206	0.347	2.297	0.400	2.381	0.321

Table A3: MSPE and standard deviation of predicted responses. From simulation with $n = 100$, $p = 1000$, $k = 3$.

A.2 With k = 15

	r = 1		r = 3		r = 5	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	0.994	0.135	1.168	0.166	1.264	0.172
Sparse	0.997	0.139	1.173	0.166	1.259	0.178
DDRP	0.985	0.136	1.145	0.171	1.221	0.167
	r = 8		r = 10		r = 15	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	1.339	0.199	1.417	0.174	1.486	0.196
Sparse	1.334	0.196	1.414	0.172	1.488	0.197
DDRP	1.272	0.191	1.339	0.185	1.377	0.184

Table A4: MSPE and standard deviation of predicted responses. From simulation with $n = 100$, $p = 110$, $k = 15$.

	r = 1		r = 3		r = 5	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	1.053	0.130	1.246	0.158	1.337	0.175
Sparse	1.055	0.129	1.246	0.162	1.337	0.173
DDRP	1.048	0.132	1.224	0.160	1.292	0.176
	r = 8		r = 10		r = 15	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	1.454	0.207	1.471	0.207	1.544	0.206
Sparse	1.455	0.208	1.468	0.207	1.540	0.202
DDRP	1.391	0.216	1.379	0.199	1.426	0.204

Table A5: MSPE and standard deviation of predicted responses. From simulation with $n = 100$, $p = 200$, $k = 15$.

	r = 1		r = 3		r = 5	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	1.059	0.148	1.305	0.167	1.394	0.214
Sparse	1.058	0.144	1.303	0.170	1.394	0.215
DDRP	1.054	0.145	1.289	0.164	1.356	0.216
	r = 8		r = 10		r = 15	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	1.537	0.228	1.579	0.239	1.665	0.215
Sparse	1.534	0.227	1.579	0.237	1.661	0.218
DDRP	1.473	0.233	1.483	0.217	1.558	0.221

Table A6: MSPE and standard deviation of predicted responses. From simulation with $n = 100$, $p = 1000$, $k = 15$.

A.3 With k = 30

	r = 1		r = 3		r = 5	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	0.660	0.099	0.746	0.114	0.793	0.110
Sparse	0.659	0.099	0.744	0.112	0.791	0.110
DDRP	0.684	0.111	0.752	0.125	0.799	0.121
	r = 8		r = 10		r = 15	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	0.810	0.120	0.859	0.117	0.870	0.117
Sparse	0.811	0.121	0.855	0.115	0.872	0.117
DDRP	0.804	0.131	0.854	0.127	0.863	0.127

Table A7: MSPE and standard deviation of predicted responses. From simulation with $n = 100$, $p = 110$, $k = 30$.

	r = 1		r = 3		r = 5	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	0.702	0.090	0.816	0.108	0.858	0.115
Sparse	0.702	0.091	0.816	0.107	0.857	0.117
DDRP	0.717	0.104	0.823	0.115	0.856	0.125
	r = 8		r = 10		r = 15	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	0.921	0.145	0.907	0.132	0.937	0.128
Sparse	0.920	0.144	0.907	0.130	0.933	0.123
DDRP	0.919	0.156	0.894	0.141	0.918	0.139

Table A8: MSPE and standard deviation of predicted responses. From simulation with $n = 100$, $p = 200$, $k = 30$.

	r = 1		r = 3		r = 5	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	0.712	0.097	0.870	0.112	0.912	0.140
Sparse	0.712	0.095	0.870	0.113	0.916	0.144
DDRP	0.717	0.100	0.872	0.114	0.917	0.147
	r = 8		r = 10		r = 15	
	MSPE	St.Dev.	MSPE	St.Dev.	MSPE	St.Dev.
Gaussian	0.995	0.153	1.003	0.143	1.061	0.147
Sparse	0.992	0.151	1.001	0.141	1.056	0.144
DDRP	0.985	0.157	0.996	0.145	1.053	0.162

Table A9: MSPE and standard deviation of predicted responses. From simulation with $n = 100$, $p = 1000$, $k = 30$.

B Derivation of β's when r and θ change

For $i = 1, ..., P$ the number of covariates, $j = 1, ..., N$ and

$$e_{ij} \sim N(0,1), \lambda_{ik} \sim N(0,1), F_{jk} \sim N(0,1)$$

independent of each other, the static factor model is given as:

$$x_{ij} = \sum_{k=1}^{r} \lambda_{ik} F_{jk} + \sqrt{\theta} e_{ij} \tag{9}$$

Define $\lambda_i \in \mathbb{R}^{r\times 1} = \begin{pmatrix} \lambda_{i1} \\ \vdots \\ \lambda_{ir} \end{pmatrix}$ and $F_j \in \mathbb{R}^{r\times 1} = \begin{pmatrix} F_{1j} \\ \vdots \\ F_{rj} \end{pmatrix}$. Then $x_{ij} = \lambda_i^\top F_j + \sqrt{\theta} e_{ij}$.

Let $x_j = \begin{pmatrix} x_{ij}, & \cdots, & x_{pj} \end{pmatrix}$ be the vector of covariates for observation j. Define $\Lambda = \begin{bmatrix} \lambda_1^\top \\ \vdots \\ \lambda_p^\top \end{bmatrix} \in \mathbb{R}^{p\times r}$ the matrix of loadings. Then $x_j = \Lambda F_j$ and $y_j = x_j^\top \beta + \eta_j$ where $\eta_j \sim N(0,1)$. We choose β by letting the signal-to-noise ratio (SNR) to one, so that the error η_j and signal $x_j^\top \beta$ have equal impact on y_j:

$$\text{SNR} = \frac{\mathbb{E}[\beta^\top x_j x_j^\top \beta]}{\text{Var}(\eta_j)} = 1$$

The expectation in the numerator then becomes:

$$\begin{aligned}
\mathbb{E}[\beta^\top x_j x_j^\top \beta] = \beta^\top \mathbb{E}[x_j x_j^\top]\beta &= \beta^\top \mathbb{E}[\Lambda F_j F_j^\top \Lambda^\top] \\
&= \mathbb{E}[(\sum_{i=1}^{r} \Lambda_{:i} F_{ji})(\sum_{i=1}^{r} \Lambda_{:i}^\top F_{ji})] \\
&= \sum_{m=1}^{r}\sum_{n=1}^{r} \mathbb{E}[\Lambda_{im} F_{jm} F_{jn} \Lambda_{in}^\top] \\
&= \sum_{i=1}^{r} \mathbb{E}[\Lambda_{:i}\Lambda_{:i}^\top \mathbb{E}[F_{ji}^2]] \\
&= \sum_{i=1}^{r} \mathbb{I}_p = r\mathbb{I}_p
\end{aligned}$$

where $\Lambda F_j = \sum_{i=1}^{r} \Lambda_{:i} F_{ji}$ and $\Lambda_{:i}$ refers to the i-th column of Λ. Then

$$\beta^\top \mathbb{E}[x_j x_j^\top]\beta = \beta_r^\top \mathbb{I}_p \beta + \beta^\top \theta \mathbb{I}_p \beta = (\beta^\top \beta)(r + \theta)$$

Let $\beta = \psi\iota$ with $\iota = \Big(1, \ \cdots, 1\Big)^\top \in \mathbb{R}^p$. Then $(\beta^\top\beta)(r+\theta) = (\psi^2\iota^\top\iota)(r+\theta) = \psi^2 p(r+\theta)$. The SNR $= 1$ implies $\psi^2 p(r+\theta) = 1$. Hence

$$\psi = \frac{1}{\sqrt{p(r+\theta}}$$

$$\implies \beta = \frac{1}{\sqrt{p(r+\theta)}}\iota$$