Fast off-the-grid sparse recovery with over-parametrized projected gradient descent

Pierre-Jean Bénard, Yann Traonmilin, Jean-François Aujol

Fast off-the-grid sparse recovery with over-parametrized projected gradient descent

Pierre-Jean Bénard[1,*], Yann Traonmilin[1], Jean-François Aujol[1]
[1] *Univ. Bordeaux, Bordeaux INP, CNRS, IMB, UMR 5251, F-33400 Talence, France*
[*] contact: pierre-jean.benard@math.u-bordeaux.fr

***Abstract*—We consider the problem of recovering off-the-grid spikes from Fourier measurements. Successful methods such as sliding Frank-Wolfe and continuous orthogonal matching pursuit (OMP) iteratively add spikes to the solution then perform a costly (when the number of spikes is large) descent on all parameters at each iteration. In 2D, it was shown that performing a projected gradient descent (PGD) from a gridded over-parametrized initialization was faster than continuous orthogonal matching pursuit. In this paper, we propose an off-the-grid over-parametrized initialization of the PGD based on OMP that permits to fully avoid grids and gives faster results in 3D.**

***Index Terms*—spike super-resolution, non-convex optimization, over-parametrization, projected gradient descent, continuous orthogonal matching pursuit**

I. Introduction

Let x_0 be an off the grid sparse signal over $\mathbb{R}^d$. Such signals can be modeled as a sum of k Dirac measures:

$$x_0 = \sum_{i=1}^{k} a_i \delta_{t_i} \tag{1}$$

where $a = (a_1, \dots, a_k) \in \mathbb{R}^k$ are the amplitudes and $t = (t_1, \dots, t_k) \in \mathbb{R}^{k \times d}$ are the locations of the spikes. We observe this signal through m Fourier measurements at frequencies $\omega = (\omega_1, \dots, \omega_m) \in \mathbb{R}^{d \times m}$. We write this as $y = Ax_0$ with A the corresponding linear operator from the space $\mathcal{M}$ of finite signed measures over $\mathbb{R}^d$ to $\mathbb{C}^m$. Note that we consider the noiseless case for the sake of clarity.

A way to recover the true signal x_0 is to find the minimizer of a non-convex least-squares problem:

$$x^* \in \underset{x \in \Sigma_{k,\epsilon}}{\arg\min} \|Ax - y\|_2^2 \tag{2}$$

where $\Sigma_{k,\epsilon}$ is a set modeling a separation constraint between spikes. Theoretical guarantees for the recovery of x_0 with (2) have been given by Gribonval and al. in [1], e.g. when frequencies are drawn with a well chosen Gaussian distribution and $m \gtrsim O(k^2 d \mathrm{polylog}(k,d))$, we have that $x^* = x_0$. Practically, continuous orthogonal matching pursuit [2] (OMP) has been successful at minimizing (2). Continuous OMP is inspired by on the the grid OMP [3] (derived from the Matching Pursuit algorithm [4]) for which theoretical success guarantees have been described [5], [6]. Success in the continuous case have been shown in cases that do not fit all practical applications (such as Dirac recovery from random Gaussian Fourier measurements) [7], [8].Another way to estimate x_0 is to add a regularization term (total variation of measures) to (2). The regularized functional can then be solved practically with the sliding Frank-Wolfe algorithm [9].

Both Continuous OMP and Sliding Frank-Wolfe iteratively add a spike then perform a descent on all parameters (amplitudes and positions). This descent step that we call the sliding step in both cases is where most of the calculations are made and it makes this class of method computationally heavy when the number of spikes is large.

The geometry of minimization (2) in the parameter space has been studied in [10]: consider a set of parameters $\Theta_{k,\epsilon}$ where

$$\begin{aligned}\Theta_{k,\epsilon} := \Big\{ &\theta = (a_1, \dots, a_k, t_1, \dots, t_k) \in \mathbb{R}^{k(d+1)}, \\ &\forall i, j \in \{1, \dots, k\}, i \neq j, \|t_i - t_j\|_2 > \epsilon \Big\}.\end{aligned} \tag{3}$$

We can rewrite (1) with the variable θ:

$$x_0 = \phi(\theta) := \sum_{i=1}^{k} a_i \delta_{t_i} \in \Sigma_{k,\epsilon}. \tag{4}$$

The unknown x_0 belongs to the low dimensional model

$$\Sigma_{k,\epsilon} := \left\{ \sum_{i=1}^{k} a_i \delta_{t_i}, (a_i, t_i) = \theta \in \Theta_{k,\epsilon} \right\}. \tag{5}$$

Problem 2 can then be equivalently written

$$\theta^* \in \underset{\theta \in \Theta_{k,\epsilon}}{\arg\min}\, g(\theta) \tag{6}$$

with

$$g(\theta) := \|A\phi(\theta) - y\|_2^2 \tag{7}$$

It has been shown that a simple gradient descent converges to a global minimimun as long as it is initialized within a basin of attraction of this minimum that has an explicit size. It has been shown that this size increases with respect to the number of measures. In [11], it was shown that a single projected gradient descent initialized by an over-parametrized backprojection of the measurements on a grid permits the recovery of a large number of spikes in 2D with improved calculation times compared to Sliding Continuous OMP. However, the use of a grid is an obstacle to the generalization of this method to domains of higher dimensions.

Contributions. In this paper, our main contribution is a fast fully grid-less method for recovering sparse signals on domains of any dimension. We propose to use an over-parametrized greedy off-the-grid initialization based on Continuous OMP without a sliding step. We then perform a

projected gradient descent that provides the final estimate. We provide experiments in 2D and 3D that shows the success of the algorithm and up to an improvement of ten times in calculation times compared to Sliding Continuous OMP. It must also be noted that with recent advances on the study of gradient descent in this context, a full proof of convergence of the proposed algorithm seems now accessible.

II. Over-parametrized Continuous OMP and Projected Gradient Descent

In this section, we describe our method which consists in two parts: a projected gradient descent and a greedy over-parametrized initialization off the grid. For each iteration of the projected Gradient Descent, we perform a heuristic where if two spikes are too close, we merge them. The Gradient Descent operates on all parameters of the estimated signal, meaning the locations and amplitudes of the spikes.

A. Projected gradient descent

The projected gradient descent (PGD), as introduced in [12], or in spectral compressed sensing in [13] and for this specific case in [11], is a way to perform a simple "descent" algorithm on the function g while guaranteeing $\theta \in \Theta_{k,\epsilon}$, i.e. our estimate of x_0 is in $\Sigma_{k,\epsilon}$.

Given an initialization $\theta_{init} \in \mathbb{R}^{k_{init}(d+1)}$, we iterate

$$\theta^{(n+1)} = P_{\Theta_\epsilon}(\theta^{(n)} - \tau_n \nabla g(\theta^{(n)})) \tag{8}$$

with τ_n the step size at the nth iteration and $P_{\Theta_{k_n,\epsilon}}$ is the projection on the separation constraint for a signal made of k_n spikes at gradient step n.

Practically, a heuristic is used to perform the projection $P_{\Theta_{k_n,\epsilon}}$ [11]: if two spikes are too close i.e. their distance from one another is below a threshold, they are merged to form one spike. To merge them, we add the amplitudes and we take the barycenter of the locations weighted by the amplitudes.

Of course, the critical step for the use of only one pass of this algorithm is that the over-parametrized initialization is close enough to θ^*. Practically, in 2D it is possible to use a simple hard thresholding of the back-projection of measurements on a grid. Given a grid Γ, we can calculate the back-projection $z_\Gamma \in \mathcal{M}$ of y on the grid Γ as

$$z_\Gamma = \sum_{s_j \in \Gamma} z_{\Gamma,j} \delta_{s_i} \quad \text{with } z_{\Gamma,j} = \sum_{l=1}^{m} y_l e^{j\langle \omega_l, s_j \rangle}. \tag{9}$$

It is then possible to extract greedily an initial θ_{init} from the largest amplitudes in z_Γ [11]. While successful in 2D for the initialization of PGD (and useful for an easy visualization of the quality of the sampling, see next section), the curse of dimension limits the possibility to extend such gridded method in domains of higher dimensions.

B. Initialization with Over-parametrized Continuous

The over-parametrized COMP (Continuous Orthogonal Matching Pursuit) is an alteration of Sliding COMP. As described in Algorithm 1, it iterates over the number of total spikes to add. It finds a location where the spike maximize a correlation with the residue. Then it performs an update of the amplitudes and it refreshes the residue. The idea is to skip the sliding part to yield an initialization close enough to the true solution for the PGD. If the norm of the residue decreases below a threshold or if the number of spikes to add passes a certain value, then we stop the iterations.

The role of over-parametrization is to compensate the inaccuracies induced at each iteration by the removal of the sliding step. This can be viewed as an off-the-grid generalization of the previously proposed gridded greedy initialization [11]. The over-parametrized COMP is described in Algorithm 1.

Algorithm 1 Continuous Orthogonal Matching Pursuit algorithm. The Sliding COMP performs state of the art spike recovery. We use an over-parametrized COMP without sliding as initialization of our PGD.

procedure COMP($A, y, K,$ `is_sliding`)
  $r^{(0)} \leftarrow y$
  $t^{(0)} \leftarrow \{\}$
  for $k = 1 \to K$ **do**
    $T \leftarrow \arg\max_t \langle A\delta_t, r^{(k-1)} \rangle$
    $t^{(k)} \leftarrow t^{(k-1)} \cup \{T\}$
    $a^{(k)} \leftarrow \arg\min_a \|A \sum_{i=1}^{k} a_i \delta_{t_i^{(k)}} - y\|_2^2$ ▷ Update of the amplitudes
    if `is_sliding` **then**
      $a^{(k)}, t^{(k)} \leftarrow$ `descent`$(g, a^{(k)}, t^{(k)})$
    end if
    $r^{(k)} \leftarrow y - A \sum_{i=1}^{k} a_i^{(k)} \delta_{t_i^{(k)}}$ ▷ Update of the residue
  end for
  return $a^{(K)}, t^{(K)}$
end procedure

To chose the amount of over-parametrization k_{init}, it was proposed to chose a fixed multiple of an estimated true number of spikes [11]. In our new method, we add spikes until it reaches a threshold on the residue or when the difference between two consecutive values of the residue reaches 0. This method makes sure that adding more spikes with our chosen COMP would not bring more information in the initialization of the projected gradient descent (as the residue would not decrease further).

It must be noted that the more robust Sliding COMP with Replacement was proposed in [14]. In the version with replacement, $2k$ spikes are estimated to produce an estimate with k spikes (making the last sliding steps more costly). We chose to compare ourselves with the faster version without replacement as the main objective of our algorithm is to provide a fast estimation (i.e. we place ourselves in the least favorable case for execution times comparison).

C. Complexity

To estimate the time gained with our method over Sliding COMP, we analyse qulaitatively the number of iterations in the descent step (which is the most time consuming) in each

method. First for the Sliding COMP method, the execution time is

$$\mathtt{T}_{SCOMP} = \mathcal{O}(N \times (1 + \cdots + K)) = \mathcal{O}(NK^2) \qquad (10)$$

with N the number of iteration in the gradient descent and K the number of spikes. For over-parametrized COMP, the execution time is

$$\mathtt{T}_{PGD} = \mathcal{O}(K \sum_i N_i C_i) \qquad (11)$$

with $N_0 = \sum_i N_i$ the number of iteration in the Projected Gradient Descent method, N_i the number of iterations between each projection and C_i the constant symbolizing the over-parameterization between each projection, i.e. after the i-th projection, there are $K_i = C_i K$ spikes left.

As the over-parametrization factors C_i are generally low compared to K, we observe that when the number of spikes K grows, the PGD becomes more interesting than Sliding COMP. We also remark PGD is faster if projections happen early in the descent.

Moreover, when compared to the grid-based initialization from [11], the over-parametrized COMP complexity is no longer exponential in the dimension of the support of the spikes.

III. Experiments

In this section, we compare our over-parametrized COMP + PGD with Sliding COMP. We study these methods in 2D and 3D in a compressed acquisition example. For both 2D and 3D cases, the signal is composed of a hundred of spikes. The domain in which the locations of the spikes are is either $[0, 1]^2$ for the 2D case or $[0, 1]^3$ for the 3D case. The code for this experiments is available for download at [15].

A. Comparisons on the 2D case

For this case, the signal has the following properties: $k = 100$, the minimum distance between two spikes is at least $\epsilon_{\text{dist}} = 0.015$. Moreover, the amplitudes follow a uniform distribution $U([1, 5])$. In addition, the number of measurements taken is $m = 40 \times k = 4000$. The frequencies of measurements follow a normal distribution $\mathcal{N}(0, c^2)$, with $c = \frac{1}{0.02} \approx \frac{1}{\epsilon_{\text{dist}}}$. The choice of the variance is primordial for the success of recovery. Indeed, The frequencies at which the signal is observed can be too high or too low resulting in a bad approximation of the signal as described in [16].

In Fig. 1(a), we represent the 2D signal as well as its back-projection. In this figure and the following, the intensity of the colors represent the amplitudes of the spikes. Some spikes close from each other form clusters that are not separated in the back-projection. For some spikes with very low amplitude (close to 1 in this case), they are barely visible on the back-projection. This means that these spikes are more complex to detect using initialization by back-projection.

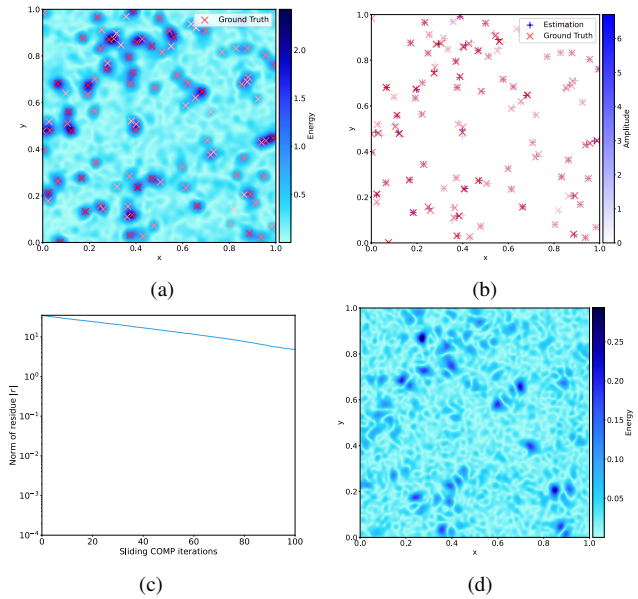


Figure 1. (a) Ground truth and its back-projection, (b) Ground truth and the estimated signal by Sliding COMP, (c) Norm of the residue after adding each spike by Sliding COMP, (d) Back-projection of the residue by Sliding COMP

1) Sliding COMP: In Fig. 1(b), Sliding COMP recovered almost all spikes with its correct amplitude. Due to the non-convex nature of the first step of COMP, some estimated spikes are stuck in a local basin of attraction. The computation time of Sliding COMP is *1h 30min*.

To compare this result to others, we use the error $e = \|y - Ax_{\text{esti}}\|_2$. We call $r := y - Ax_{\text{esti}}$ the residue.

As we initialize the estimated signal spike by spike, the norm of the residue r decreases steadily to attain $\|r^{(k)}\|_2 \approx 5$. As we see in Fig. 1(c), the norm of the residue is still decreasing at the end, meaning that the optimization and/or the adding process has not completely converged to the true solution. However, if we compare the back-projection of the observed signal y in Fig. 1(a) with the back-projection of the residue $r^{(k)}$ as in Fig. 1(d), the order of magnitude of the energy has decreased. This means that Sliding COMP recovered the majority of the signal.

2) Over-parametrized COMP without PGD : To understand the role of the initialization with the over-parametrized COMP, we show its results without the PGD step in Fig. 2(a). We observe that every spikes of the true signal has at least an initialized spike close to it. In some places, we notice a cluster of spikes. This is for these cases that the projection part is needed. The time to compute the initialization is *1min 54s*. We note the low cost of this initialization compared to Sliding COMP. We see that in Fig. 2(b), the norm of the residue is steadily decreasing and that in the last few steps, it stays constant. This is the limit that this method attains and adding more spikes does not increases the accuracy of the estimation.

By comparing the back-projection of the residue after initialization with the back-projection of the observation, we see in Fig. 2(c) that the order of magnitude of the error has decreased. Yet there are still spots with a lot of energy meaning that the estimation is not accurate enough. This is why we still

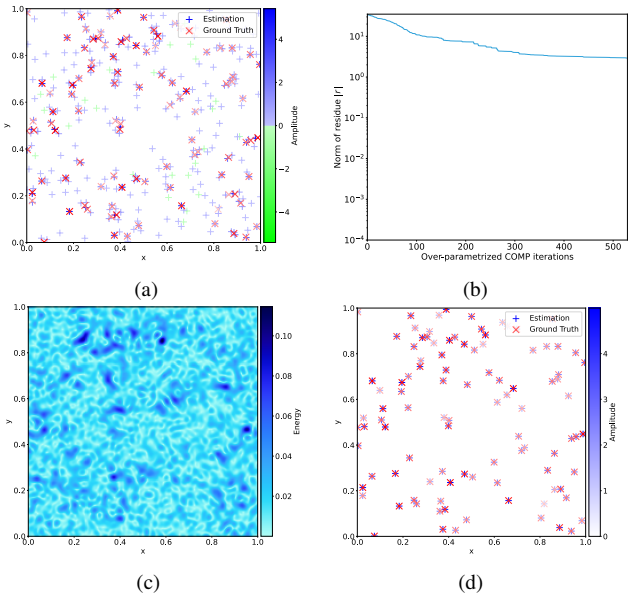


Figure 2. (a) Ground truth and initialized signal by over-parametrized COMP, (b) Norm of the residue after adding each spike by over-parametrized COMP, (c) Back-projection of the initialized residue by over-parametrized COMP, (d) Ground truth and estimated signal by over-parametrized COMP

need the PGD to optimize the initialized signal.

3) Over-parametrized COMP with PGD : After the optimization step with the PGD, we get the following result in Fig. 2(d). All the spikes from the ground truth have been estimated. This procedure took approximately *24min* and it converged after 2656 iterations of the PGD. We consider that an estimated signal has converged to the ground truth if the norm of the residue is lower that a threshold. With a much smaller calculation time our method was able to recover more accurately the spikes. Improvement in calculation times would be even greater when comparing to the more accurate but slower Sliding COMP with replacement.

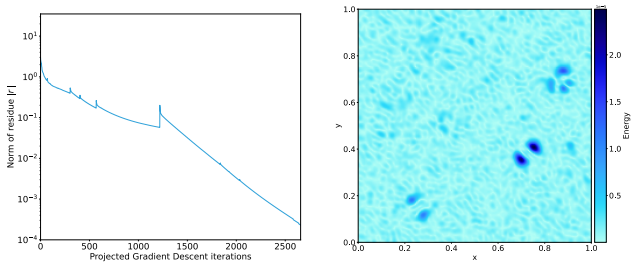


Figure 3. (a) Norm of the residue during the PGD, (b) Back-projection of the estimated residue by PGD

By observing Fig. 3(a), we also deduce that all the signal has been estimated. We note that the norm of the residue may increase sometimes during the PGD. This is a typical phenomenon that must be controlled to bre able to proove convergence. Finally, the norm of the residue is close to zero and the back-projection of the residue has an energy very low compared to the original back-projection of the observation of the true signal in Fig. 3(b). We can note the difference of scale with the back-projection of the residue obtained with Sliding COMP.

4) Limits of the over-parametrization: Computing the amplitudes at each step is done by solving a least-squares problem with a matrix M of size $m \times k$. So the more spikes we add, the bigger M gets. We show that in Figs. 4(a) and 4(b), the condition number of M at the end of the over-parametrized COMP is higher than for Sliding COMP. This means that too much over-parametrization lead to a condition number so high that it can induce errors in the computation. This also shows that the amount of over-parametrization by COMP cannot be arbitrarily large.

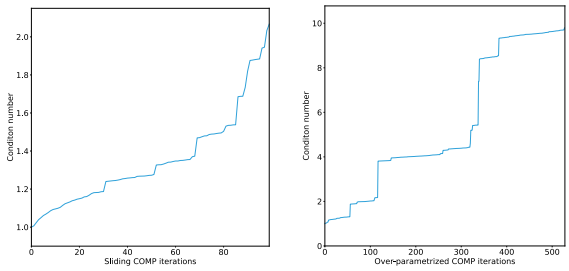


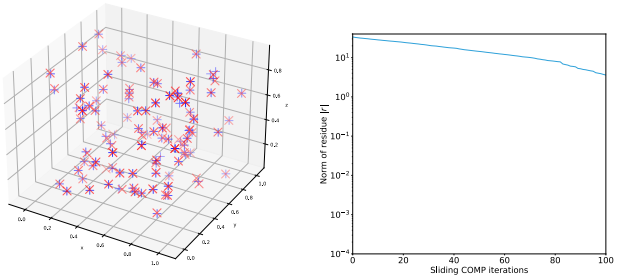
Figure 4. (a) Condition number of M by Sliding COMP, (b) Condition number of M by over-parametrized COMP

B. Comparisons on the 3D case

For the the recovery of a signal in 3D (represented in red in Fig. 5(a)), we perform the same analysis. The same parameters than the 2D case are applied to some exceptions which are the following. The minimum separation between spikes is set to $\epsilon_{\text{dist}} = 0.05$. Furthermore, since the measurements depend in our case on ϵ_{dist}, they follow a normal distribution $\mathcal{N}(0, c^2)$ with $c = \frac{1}{0.05} \approx \frac{1}{\epsilon_{\text{dist}}}$.

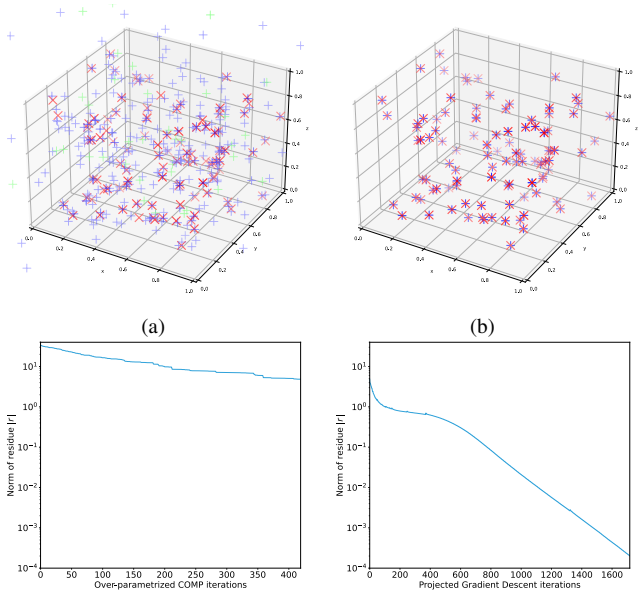
1) Sliding COMP: We see that in Fig. 5(a), most spikes are also recovered with some exceptions. It is important to note that in some place, two or more spikes from the ground truth signal have been estimated by a single spike of the estimated signal. The amplitude of this spike is roughly the sum of the amplitudes of the spikes in the cluster. Sliding COMP computed this estimation in approximately *1h 30min*. For the norm of the residue in Fig. 5(b), it is decreasing over the iterations but does not attain 0 after adding 100 spikes. We also note that this time is not too far apart from the time in the 2D case. We can deduce that this family of algorithm is not too dependent on the dimension of the space. On the contrary to method needing a grid like in [11].

2) Over-parametrized COMP + PGD: : After the initialization, we get the signal in Fig. 6(a). We note that all the spikes from the ground truth have at least one spike from the initialized signal. Moreover, some spikes have a negative amplitude (in green). This step took approximately *1min* after adding 418 spikes. In Fig. 6(c), we note that the norm of the residue decreases until it reaches a threshold. Adding even more spikes does not increase the accuracy of the estimation.



(a) (b)

Figure 5. (a) Ground truth and estimation by Sliding COMP, (b) Norm of the residue during the Sliding COMP



(a) (b)

(c) (d)

Figure 6. (a) Ground truth and initialization by Sliding COMP, (b) Ground truth and estimation by PGD, (c) Norm of the residue during the over-parametrized COMP, (d) Norm of the residue during the PGD

Same as Sliding COMP, this algorithm seems not to be too dependent from the number of dimensions.

For the projected gradient descent, we note that after 1716 iterations, the energy of the system reached a threshold close to 0 to consider convergence in Fig. 6(d). It took approximately *13min 45s*. We get as a final result the signal in Fig. 6(b).

IV. Discussion/Conclusion

We showed that Projected Gradient Descent initialized by over-parametrized COMP without sliding leads to better results in faster times than Sliding COMP. Indeed, it provides a way to catch all spikes from the ground truth and without needing to know precisely the number of true spikes to recover. We have shown the success of our method in both 2D and 3D and expect similar results for examples in larges dimensions. To recapitulate, the table I shows the computation times of each algorithms.

Although these are some promising experimental results, there are still no theoretical guarantees for this new method. However, some very strong quantitative and qualitative insights already exist [10], [17], and hope for the possible extension of proofs from the finite dimensional domain to our case gives interesting potential leads for future work.

Table I
Summary of computation times

	Time (in min.)	
Dimensions	2D	3D
Sliding COMP	90	90
Over-parametrized COMP	2	1
Projected Gradient Descent	24	14

V. Acknowledgments

This work was supported by the French National Research Agency (ANR) under reference ANR-20-CE40-0001 (EFFIREG project).

References

[1] R. Gribonval, G. Blanchard, N. Keriven, and Y. Traonmilin, "Compressive Statistical Learning with Random Feature Moments," *Mathematical Statistics and Learning*, vol. 3, no. 2, pp. 113–164, Aug. 2021.

[2] N. Keriven, N. Tremblay, Y. Traonmilin, and R. Gribonval, "Compressive k-means," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 6369–6373.

[3] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proceedings of 27th Asilomar conference on signals, systems and computers*. IEEE, 1993, pp. 40–44.

[4] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on signal processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[5] J. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[6] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on information theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[7] C. Elvira, R. Gribonval, C. Soussen, and C. Herzet, "OMP and continuous dictionaries: Is k-step recovery possible?" in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5546–5550.

[8] ——, "When does OMP achieve exact recovery with continuous dictionaries?" *Applied and Computational Harmonic Analysis*, vol. 51, p. 39, Mar. 2021.

[9] Q. Denoyelle, V. Duval, G. Peyré, and E. Soubies, "The Sliding Frank-Wolfe Algorithm and its Application to Super-Resolution Microscopy," *Inverse Problems*, 2019.

[10] Y. Traonmilin and J.-F. Aujol, "The basins of attraction of the global minimizers of the non-convex sparse spike estimation problem," *Inverse Problems*, Feb. 2020.

[11] Y. Traonmilin, J.-F. Aujol, and A. Leclaire, "Projected gradient descent for non-convex sparse spike estimation," *IEEE Signal Processing Letters*, vol. 27, pp. 1110–1114, 2020.

[12] Y. Chen and M. J. Wainwright, "Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees," 2015.

[13] J.-F. Cai, T. Wang, and K. Wei, "Spectral compressed sensing via projected gradient descent," *SIAM Journal on Optimization*, vol. 28, no. 3, pp. 2625–2653, 2018.

[14] N. Keriven and R. Gribonval, "Compressive Gaussian Mixture Estimation by Orthogonal Matching Pursuit with Replacement," Jul. 2015.

[15] J.-F. A. Pierre-Jean Bénard, Yann Traonmilin. (2022, Feb.) Code of the experiments. [Online]. Available: https://github.com/pjbenard/opCOMP_sparse_recovery

[16] A. Chatalic and R. Gribonval, "Learning to sketch for compresssive clustering," in *iTWIST*, June 2020.

[17] Y. Traonmilin, J. Aujol, and A. Leclaire, "The basins of attraction of the global minimizers of non-convex inverse problems with low-dimensional models in infinite dimension," *CoRR*, vol. abs/2009.08670, 2020.