SUPPLEMENTARY MATERIALS FOR "ON SUPER-RESOLUTION WITH SEPARATION PRIOR"

Xingyun Mao and Heng Qiao

University of Michigan - Shanghai Jiao Tong University Joint Institute,
Shanghai Jiao Tong University, Shanghai, China
E-mail: mxy123@sjtu.edu.cn, heng.qiao@sjtu.edu.cn

In this file, we provide all the proofs, complete algorithms and examples in companion with the main body paper.

1. ONE-DIMENSIONAL MAPPING OPERATOR

$$\mathcal{P}_{\Delta}^{k,M}(\{\boldsymbol{\mu}_i\}_{i=1}^k) \in \arg\min_{\{\hat{\boldsymbol{\mu}}_i\}_{i=1}^k \subset \mathbb{R}^M} d_p(\{\boldsymbol{\mu}_i\}_{i=1}^k, \{\hat{\boldsymbol{\mu}}_i\}_{i=1}^k)$$
$$s.t. \quad \|\hat{\boldsymbol{\mu}}_i - \hat{\boldsymbol{\mu}}_j\|_2 \geqslant \Delta, \quad \forall 1 \leqslant i \neq j \leqslant k \tag{1}$$

Theorem 1. *The on-grid mapping $\tilde{\mathcal{P}}_{\Delta,\delta}^{k,1}$ in Algorithm 2 is optimal in the sense of (1) when both $\boldsymbol{\mu}, \hat{\boldsymbol{\mu}} \in \mathbb{R}^k$ are restricted on the grid. With grid size δ, the number of arithmetic operations of $\tilde{\mathcal{P}}_{\Delta,\delta}^{k,1}$ is upper bounded by $3k(k-1)\lceil \frac{\Delta}{\delta} \rceil$.*

Proof. **(Main Idea of Algorithm 2)**

In Algorithm 2, from line 1 to 5, we first set the range of the result that $\hat{t}_i \in [g_{\min}, g_{\max}]$. Then each t_i is normalized as $q_i \triangleq (t_i - g_{\min})/\delta \in \mathbb{N}$. We denote the total number of grids to be considered as N, $q_i \in [0, N-1]$, and convert Δ-separated constraint ($|t_i - t_j| \geqslant \Delta, \ \forall i \neq j$) into s-grid-separation ($|q_i - q_j| \geqslant s$).

Afterwards, we find one projection $\hat{\boldsymbol{q}}$ such that

$$\hat{\boldsymbol{q}} = \arg\min_{\hat{\boldsymbol{q}}} d_p(\boldsymbol{q}, \hat{\boldsymbol{q}}) \quad \text{s.t.} \quad \hat{q}_{i+1} - \hat{q}_i \geqslant s, \ i = 1, ..., k-1$$

$\hat{\boldsymbol{q}}$ may not be unique, and the algorithm obtains one of them. Then the corresponding $\hat{\boldsymbol{t}}$ is recovered. Since $\boldsymbol{q}$ and $\hat{\boldsymbol{q}}$ are normalizations of $\boldsymbol{t}$ and $\hat{\boldsymbol{t}}$, the optimality in Theorem 1 can be proved, if s-separated $\boldsymbol{q}, \hat{\boldsymbol{q}} \in \mathbb{N}^k$ are optimal in the sense of (1), which is shown below.

(Optimality for On-Grid Mapping)

The dynamic program determines an auxiliary matrix $\mathbf{F}(k, N)$, where N is the number of grids we consider and $\mathbf{F}(r, n)$ is defined for $1 \leqslant r \leqslant k$ and $0 \leqslant n \leqslant N-1$ by

$$\mathbf{F}(r,n) := \min \left\{ \sum_{i=1}^{r} |q_i - \hat{q}_i|^p \ \middle| \ \hat{q}_r \leqslant n \right\} \tag{2}$$

We claim that, for $2 \leqslant r \leqslant k$ and $s(r-1) \leqslant n \leqslant N-1$,

$$\mathbf{F}(r,n) = \min \begin{cases} \mathbf{F}(r-1, n-s) + |q_r - n|^p, \\ \mathbf{F}(r, n-1). \end{cases} \tag{3}$$

The relation is straightforward, as it distinguishes whether the last entry of the minimizer for $\mathbf{F}(r,n)$ is equal to or less than n. To be specific, we establish the estimate on $\mathbf{F}(r,n)$ by considering the minimizer $\hat{\boldsymbol{q}}_{[1:r]} \in \mathbb{R}^r$ for $\mathbf{F}(r,n)$: if $\hat{q}_r < n$, $\hat{q}_r \leqslant n-1$, so $\mathbf{F}(r,n) = \mathbf{F}(r, n-1)$ by the definition (2); if $\hat{q}_r = n$, so that $\hat{q}_{r-1} \leqslant n-s$, then $\mathbf{F}(r,n) = \sum_{i=1}^{r-1} |q_i - \hat{q}_i|^p + |q_r - \hat{q}_r|^p = \mathbf{F}(r-1, n-s) + |q_r - n|^p$. With the relation (2) now fully justified, table $\mathbf{F}$ can be filled with initial values

$$\mathbf{F}(1,n) = \min \begin{cases} |q_1 - n|^p, & \text{if } n < q_1 \\ 0, & \text{otherwise.} \end{cases}$$
$$\mathbf{F}(r,n) = \infty, \quad 2 \leqslant r \leqslant k, \quad 0 \leqslant n < s(r-1)$$

In the former relation where $r = 1$, if $n \geqslant q_1$, we have $\hat{q}_1 = q_1$, so $\mathbf{F}(1,n) = 0$; if $n < q_1$, the minimizer is $\hat{q}_1 = n$, so $\mathbf{F}(1,n) = |q_1 - n|^p$. The latter relation reflects the non existence of r-length s-separated vector within $s(r-1)$ grids.

(Complexity)

According to (2), $\mathbf{F}$ contains kN entries. Since only the optimal projections do matter, it is not necessary to compute all entries. Specifically, when determining $\mathbf{F}(r, \cdot)$s for a fixed r, we narrow the range of optimal $\hat{q}_r$ to $[q_r - (k-r)s, q_r + (r-1)s]$ instead of $[0, N-1]$. The reason is that if $\hat{q}_r > q_r + (r-1)s$, then we can construct s-separated $\hat{\boldsymbol{q}}'$ with $\hat{q}'_i = \min(q_r + (i-1)s, \hat{q}_i)$ for $i = 1, ..., r$ and $\hat{q}'_j = \hat{q}_j$ for $j = r+1, ..., k$, which satisfies $d_p(\hat{\boldsymbol{q}}', \boldsymbol{q}) < d_p(\hat{\boldsymbol{q}}, \boldsymbol{q})$. Similarly, if $\hat{q}_r < q_r - (k-r)s$, then we can construct s-separated $\hat{\boldsymbol{q}}'$ with $\hat{q}'_i = \hat{q}_i$ for $i = 1, ..., r$ and $\hat{q}'_j = \max(q_r - (j-r)s, \hat{q}_j)$ for $j = r+1, ..., k$, which satisfies $d_p(\hat{\boldsymbol{q}}', \boldsymbol{q}) < d_p(\hat{\boldsymbol{q}}, \boldsymbol{q})$.

Hence, we can replace the conditions in line 7 and 15 in Algorithm 2 and get Algorithm 3. In this way, for each r, at most $(k-1)s$ entries of $\mathbf{F}$ will be computed with each entry requiring three basic arithmetic operations in (3) - one addition, one subtraction and one exponentiation. The total complexity is upper bounded by $3k(k-1)\lceil \frac{\Delta}{\delta} \rceil$ arithmetic operations. When k is large, this is an improvement compared with $\mathcal{O}(k^{3.5} \log(\frac{1}{\delta}))$ for the quadratic programming strategy of [1].

(Example)

As for the best projection itself, we need to back track the case producing $\mathbf{F}$. The process is fully specified in Algorithm 4. Table 1 displays an example of $\boldsymbol{q} = [6, 7, 8, 9]^T$, $s = 2$ and $p = 2$ with corresponding $\mathbf{F}$, where one of the best projections is $\hat{\boldsymbol{q}} = [4, 6, 8, 10]^T$. To acquire one best projection, we follow the path of arrows starting from the $(k, N-1)$th box until $r = 1$: if an arrow points northwest from the (r, n)th box, then the grid n is selected for the entry of the best approximation, and if $r = 1$, the pointed grid is selected. Once one best projection $\hat{\boldsymbol{q}}$ is determined in this way, its corresponding $\hat{\boldsymbol{t}}$ is recovered as return value.

$\boldsymbol{q}$	$\mathbf{F}(r,n)$	$r=1$	$r=2$	$r=3$	$r=4$
-	$n=0$	36	⋆	⋆	⋆
-	$n=1$	25	⋆	⋆	⋆
-	$n=2$	16	⋆	⋆	⋆
-	$n=3$	9	41	⋆	⋆
-	$n=4$	4	25	⋆	⋆
$q_1=6$	$n=5$	1	13	⋆	⋆
$q_2=7$	$n=6$	0	5	29	⋆
$q_3=8$	$n=7$	⋆	1	14	⋆
$q_4=9$	$n=8$	⋆	1	5	⋆
-	$n=9$	⋆	1	2	14
-	$n=10$	⋆	⋆	2	6
-	$n=11$	⋆	⋆	2	6
-	$n=12$	⋆	⋆	2	6
-	$n=13$	⋆	⋆	⋆	6
-	$n=14$	⋆	⋆	⋆	6
-	$n=15$	⋆	⋆	⋆	6

Table 1: Sketch of the dynamic program computing the best s-separated projections of $\boldsymbol{q} = [6, 7, 8, 9]^T$ with $s = 2$ and $p = 2$. "⋆" means the entry is not required to be considered.

□

Lemma 1. *The one-dimensional transportation distance in ℓ_p norm ($p \geqslant 1$) satisfies triangle inequalities as follows*

$$d_p^{1/p}(\boldsymbol{a}, \boldsymbol{b}) + d_p^{1/p}(\boldsymbol{b}, \boldsymbol{c}) \geqslant d_p^{1/p}(\boldsymbol{a}, \boldsymbol{c})$$

where $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^k$ are sorted.

Proof. When $\boldsymbol{a}, \boldsymbol{b}$ are sorted, $d_p(\boldsymbol{a}, \boldsymbol{b}) = \sum_{i=1}^k |a_i - b_i|^p = ||\boldsymbol{a} - \boldsymbol{b}||_p^p$. The relation can be acquired through Minkowski inequality

$$\begin{aligned} d_p^{1/p}(\boldsymbol{a}, \boldsymbol{b}) + d_p^{1/p}(\boldsymbol{b}, \boldsymbol{c}) &= ||\boldsymbol{a} - \boldsymbol{b}||_p + ||\boldsymbol{b} - \boldsymbol{c}||_p \\ &\geqslant ||\boldsymbol{a} - \boldsymbol{c}||_p = d_p^{1/p}(\boldsymbol{a}, \boldsymbol{c}) \end{aligned}$$

□

Theorem 2. *Let $\boldsymbol{\mu}^\star$ be one gridless optimal solution of (1) and $\boldsymbol{\mu}^\# = \mathcal{P}_\Delta^{k,1}(\boldsymbol{\mu})$. If $\frac{\Delta}{\delta} = m \in \mathbb{N} = \{0, 1, \cdots\}$, then we have*

$$d_p^{1/p}(\boldsymbol{\mu}^\#, \boldsymbol{\mu}) \leqslant d_p^{1/p}(\boldsymbol{\mu}^\star, \boldsymbol{\mu}) + 1.5\delta k^{1/p}$$

Proof. We obtain the on-grid approximations $\boldsymbol{t}$ and $\boldsymbol{t}^\star$ of $\boldsymbol{\mu}$ and $\boldsymbol{\mu}^\star$ respectively as Step 1 in Algorithm 1. Let $\mu_i = t_i + \epsilon_i$ and $\mu_i^\star = t_i^\star + \epsilon_i^\star$. It is obvious that $-0.5\delta \leqslant \epsilon_i, \epsilon_i^\star < 0.5\delta$ for $i = 1, ..., k$, which implies $d_p(\boldsymbol{\mu}, \boldsymbol{t}), d_p(\boldsymbol{\mu}^\star, \boldsymbol{t}^\star) \leqslant (0.5\delta)^p k$.

As the ground-truth $\boldsymbol{\mu}^\star$ is Δ-separated, we have

$$\begin{aligned} |\mu_{i+1}^\star - \mu_i^\star| &= |t_{i+1}^\star - t_i^\star + \epsilon_{i+1}^\star - \epsilon_i^\star| \\ &\leqslant |t_{i+1}^\star - t_i^\star| + |\epsilon_{i+1}^\star - \epsilon_i^\star| \\ &< |t_{i+1}^\star - t_i^\star| + \delta \end{aligned}$$

which indicates $|t_{i+1}^\star - t_i^\star| > \Delta - \delta = (m-1)\delta$. Since $\boldsymbol{t}^\star$ is on grid and $\Delta/\delta \in \mathbb{N}$, we must have $|t_{i+1}^\star - t_i^\star| \geqslant m\delta = \Delta$. Next, we can apply the triangle inequality and have

$$\begin{aligned} d_p^{1/p}(\boldsymbol{\mu}, \boldsymbol{\mu}^\star) &\geqslant d_p^{1/p}(\boldsymbol{\mu}, \boldsymbol{t}^\star) - d_p^{1/p}(\boldsymbol{\mu}^\star, \boldsymbol{t}^\star) \\ &\geqslant d_p^{1/p}(\boldsymbol{t}, \boldsymbol{t}^\star) - d_p^{1/p}(\boldsymbol{\mu}, \boldsymbol{t}) - d_p^{1/p}(\boldsymbol{\mu}^\star, \boldsymbol{t}^\star) \\ &\geqslant d_p^{1/p}(\boldsymbol{t}, \boldsymbol{\mu}^\#) - d_p^{1/p}(\boldsymbol{\mu}, \boldsymbol{t}) - d_p^{1/p}(\boldsymbol{\mu}^\star, \boldsymbol{t}^\star) \\ &\geqslant d_p^{1/p}(\boldsymbol{\mu}, \boldsymbol{\mu}^\#) - 2d_p^{1/p}(\boldsymbol{\mu}, \boldsymbol{t}) - d_p^{1/p}(\boldsymbol{\mu}^\star, \boldsymbol{t}^\star) \\ &\geqslant d_p^{1/p}(\boldsymbol{\mu}, \boldsymbol{\mu}^\#) - 1.5\delta k^{1/p} \end{aligned}$$

where the first, second and fourth steps use the inequality in Lemma 1 and the third step uses the optimality in Theorem 1 that $d_p(\boldsymbol{t}^\star, \boldsymbol{t}) \geqslant d_p(\boldsymbol{\mu}^\#, \boldsymbol{t})$.

□

Algorithm 1 One-Dimensional Mapping Operator $\mathcal{P}_\Delta^{k,1}$

Input: grids $\mathcal{G} = \{g \pm i\delta \mid i \in \mathbb{Z},\ g, \delta \in \mathbb{R}\}$; $\boldsymbol{\mu} \in \mathbb{R}^k$, Δ;
Output: Δ-separated approximation $\hat{\boldsymbol{\mu}}$ on $\mathcal{G}$;
1: $\boldsymbol{t} \leftarrow \arg\min_{g_i \in \mathcal{G}} ||\boldsymbol{g} - \boldsymbol{\mu}||_2$
2: $\hat{\boldsymbol{\mu}} \leftarrow \tilde{\mathcal{P}}_{\Delta,\delta}^{k,1}(\boldsymbol{t})$
3: **return** $\hat{\boldsymbol{\mu}}$

2. TWO DIMENSIONAL MAPPING OPERATOR

Theorem 3. *The output $\{\hat{\boldsymbol{\mu}}_i\}_{i=1}^k$ in Algorithm 5 is Δ-separated and there exists $\pi \in \Pi_k$ such that*

$$\sum_{i=1}^k \|\boldsymbol{\mu}_{\pi(i)} - \hat{\boldsymbol{\mu}}_i\|_2 \leqslant \frac{k(k-1)\Delta}{2} \quad (4)$$

$$\max_{1 \leqslant i \leqslant k} \|\boldsymbol{\mu}_{\pi(i)} - \hat{\boldsymbol{\mu}}_i\|_2 \leqslant (k-1)\Delta \quad (5)$$

Proof. **(Main Idea of Algorithm 5)**

In Algorithm 5, all $\{\boldsymbol{\mu}_i\}_{i=1}^k$ are moved away from $\boldsymbol{c}$ iteratively, such that the distance between each two points will increase and exceed Δ. Specifically, at the i-th iteration, we set $\hat{\boldsymbol{\mu}}_i$ as the closest point to $\boldsymbol{c}$ among $\{\hat{\boldsymbol{\mu}}_l\}_{l=i}^k$ and fix it while moving each $\hat{\boldsymbol{\mu}}_j$ ($j > i$) away from $\boldsymbol{c}$ to ensure $||\hat{\boldsymbol{\mu}}_i - \hat{\boldsymbol{\mu}}_j||_2 \geqslant \Delta$ for all $j > i$. One example with $k = 5$ and $\Delta = 1$ is given in Fig. 1a.

Algorithm 2 On-Grid Mapping Operator $\tilde{\mathcal{P}}^{k,1}_{\Delta,\delta}$

Input: $\boldsymbol{t} \in \mathbb{R}^k$
Output: Δ-separated approximation $\hat{\boldsymbol{t}}$ on $\mathcal{G}$;
$s \leftarrow \lceil \Delta/\delta \rceil$
$g_{\min} \leftarrow t_1 - (k-1)s\delta$
$g_{\max} \leftarrow t_k + (k-1)s\delta$
$\boldsymbol{q} \leftarrow (\boldsymbol{t} - g_{\min})/\delta$
$N \leftarrow (g_{\max} - g_{\min})/\delta + 1$
$\mathbf{F} \leftarrow$ matrix with size $k \times N$ and all elements equal to ∞
for $n = 0 \rightarrow N-1$ **do**
  if $n < q_1$ **then**
    $\mathbf{F}(1,n) \leftarrow |q_1 - n|^p$
  else
    $\mathbf{F}(1,n) \leftarrow 0$
  end if
end for
for $r = 2 \rightarrow k$ **do**
  for $n = s(r-1) \rightarrow N-1$ **do**
    $\mathbf{F}(r,n) \leftarrow \min(\mathbf{F}(r-1,n-s) + |q_r - n|^p,$
          $\mathbf{F}(r,n-1))$
  end for
end for
$\hat{\boldsymbol{q}} \leftarrow$ BackTracking($\mathbf{F}$)
$\hat{\boldsymbol{t}} \leftarrow \hat{\boldsymbol{q}}\delta + g_{\min}$
return $\hat{\boldsymbol{t}}$

For the moving distance of $\hat{\boldsymbol{\mu}}_j$ at the i-th iteration ($j > i$), we take it as small as possible, which means only if $||\hat{\boldsymbol{\mu}}_i - \hat{\boldsymbol{\mu}}_j||_2 < \Delta$, $\hat{\boldsymbol{\mu}}_j$ is moved along the direction $(\hat{\boldsymbol{\mu}}_j - \boldsymbol{c})$ until $||\hat{\boldsymbol{\mu}}_i - \hat{\boldsymbol{\mu}}_j||_2 = \Delta$. The position of $\hat{\boldsymbol{\mu}}_j$ after moving is determined with cosine theorem, as is shown in Fig. 1b. Denote that $\hat{\boldsymbol{\mu}}_j$ is moved to $\hat{\boldsymbol{\mu}}'_j$. We have

$$\begin{aligned}\cos(\boldsymbol{\varphi}) &= \frac{||\hat{\boldsymbol{\mu}}_i - \boldsymbol{c}||_2^2 + ||\hat{\boldsymbol{\mu}}_j - \boldsymbol{c}||_2^2 - ||\hat{\boldsymbol{\mu}}_i - \hat{\boldsymbol{\mu}}_j||_2^2}{2||\hat{\boldsymbol{\mu}}_i - \boldsymbol{c}||_2||\hat{\boldsymbol{\mu}}_j - \boldsymbol{c}||_2} \\ &= \frac{||\hat{\boldsymbol{\mu}}_i - \boldsymbol{c}||_2^2 + ||\hat{\boldsymbol{\mu}}'_j - \boldsymbol{c}||_2^2 - ||\hat{\boldsymbol{\mu}}_i - \hat{\boldsymbol{\mu}}'_j||_2^2}{2||\hat{\boldsymbol{\mu}}_i - \boldsymbol{c}||_2||\hat{\boldsymbol{\mu}}'_j - \boldsymbol{c}||_2}\end{aligned}$$

where $||\hat{\boldsymbol{\mu}}_i - \hat{\boldsymbol{\mu}}'_j||^2 = \Delta$. The equation is simplified as line 6 and 7 in Algorithm 5.

(Validity of output and Proof of (4) and (5))

The i-th iteration will guarantee that $\|\hat{\boldsymbol{\mu}}_i - \hat{\boldsymbol{\mu}}_j\|_2 \geqslant \Delta$ for all $j > i$. To prove the validity of Algorithm 5, we need to show that any l-th iteration with $l > i$ will still ensure $||\hat{\boldsymbol{\mu}}_i - \hat{\boldsymbol{\mu}}_j||_2 \geqslant \Delta$ for $j > i$. Indeed, as illustrated in Fig. 1(b), assume that $\hat{\boldsymbol{\mu}}_j$ is moved to $\hat{\boldsymbol{\mu}}'_j$ at some l-th iteration for $j > l > i$, we have $\|\hat{\boldsymbol{\mu}}_i - \boldsymbol{c}||_2 \leqslant ||\hat{\boldsymbol{\mu}}_j - \boldsymbol{c}\|_2$ which implies $\angle\beta < \frac{\pi}{2}$ and $\|\hat{\boldsymbol{\mu}}_i - \hat{\boldsymbol{\mu}}'_j\|_2 > \|\hat{\boldsymbol{\mu}}_i - \hat{\boldsymbol{\mu}}_j\|_2 \geqslant \Delta$ as expected.

As for (4) and (5), we construct π by setting $\pi(i) = i$ for $i = 1, ..., k$ at first and swapping $\pi(i)$ and $\pi(l)$ at i-th iteration according to line 3 in Algorithm 5 to ensure one-to-one correspondence of $\{\boldsymbol{\mu}_i\}_{i=1}^k$ and $\{\hat{\boldsymbol{\mu}}_i\}_{i=1}^k$.

Algorithm 3 On-grid Mapping Operator $\tilde{\mathcal{P}}^{k,1}_{\Delta,\delta}$

Input: $\boldsymbol{t} \in \mathbb{R}^k$
Output: Δ-separated approximation $\hat{\boldsymbol{t}}$ on $\mathcal{G}$;
$s \leftarrow \lceil \Delta/\delta \rceil$
$g_{\min} \leftarrow t_1 - (k-1)s\delta$
$g_{\max} \leftarrow t_k + (k-1)s\delta$
$\boldsymbol{q} \leftarrow (\boldsymbol{t} - g_{\min})/\delta$
$N \leftarrow (g_{\max} - g_{\min})/\delta + 1$
$\mathbf{F} \leftarrow$ matrix with size $k \times N$ and all elements equal to ∞
for $n = 0 \rightarrow q_1$ **do**
  $\mathbf{F}(1,n) \leftarrow |q_1 - n|^p$
end for
for $r = 2 \rightarrow k$ **do**
  for $n = \max(s(r-1), q_r - (k-r)s) \rightarrow q_r + (r-1)s$ **do**
    $\mathbf{F}(r,n) \leftarrow \min(\mathbf{F}(r-1,n-s) + |q_r - n|^p,$
          $\mathbf{F}(r,n-1))$
  end for
end for
$\hat{\boldsymbol{q}} \leftarrow$ BackTracking($\mathbf{F}$)
$\hat{\boldsymbol{t}} \leftarrow \hat{\boldsymbol{q}}\delta + g_{\min}$
return $\hat{\boldsymbol{t}}$

In this way, the left hand side of (4) is the total sum of moving distance. Since there are at most $k(k-1)$ movements of points with each movement distance $||\hat{\boldsymbol{\mu}}'_j - \hat{\boldsymbol{\mu}}_j||_2 \leqslant \Delta$ as in Fig.1b, the total sum is no larger than $\frac{k(k-1)\Delta}{2}$. Similarly, the (5) follows from the fact that any given point $\hat{\boldsymbol{\mu}}$ can be moved at most $k-1$ times, so its maximum movement distance is $(k-1)\Delta$.

(Extreme instance of (4) and (5))

Let $\{\boldsymbol{\mu}_i\}_{i=1}^k = \{[(i-1)\epsilon, 0]\}_{i=1}^k$ for $\epsilon > 0$ and $\boldsymbol{c} = [0,0]$. According to Algorithm 5, the output $\{\hat{\boldsymbol{\mu}}_i\}_{i=1}^k = \{[(i-1)\Delta, 0]\}_{i=1}^k$, which implies $\sum_{i=1}^k \|\boldsymbol{\mu}_i - \hat{\boldsymbol{\mu}}_i\|_2 = \frac{k(k-1)(\Delta-\epsilon)}{2} < \frac{k(k-1)\Delta}{2}$. Similarly, $\|\boldsymbol{\mu}_k - \hat{\boldsymbol{\mu}}_k\|_2 = (k-1)(\Delta - \epsilon) < (k-1)\Delta$. As $\epsilon \rightarrow 0$, both bounds are nearly

Algorithm 4 BackTracking

Input: $\mathbf{F} \in \mathbb{R}^{k \times N}$
Output: $\hat{\boldsymbol{q}} \in \mathbb{R}^k$;
$r \leftarrow k$; $n \leftarrow N-1$
while $r > 0$ **do**
  if $\mathbf{F}(r,n) = \mathbf{F}(r,n-1)$ **then**
    $n \leftarrow n-1$
  else
    $\hat{q}_r \leftarrow n$
    $n \leftarrow n-s$; $r \leftarrow r-1$
  end if
end while
return $\hat{\boldsymbol{q}}$

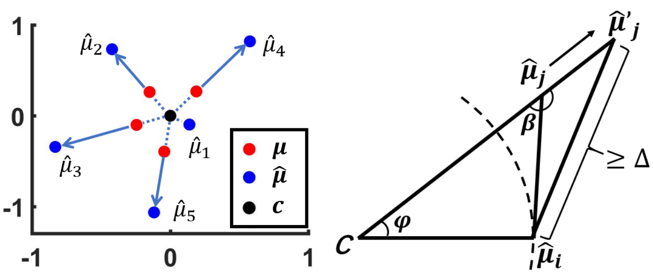
Algorithm 5 Two-Dimensional Mapping Operator $\mathcal{P}_{\Delta}^{k,2}$

Input: $\{\boldsymbol{\mu}_i\}_{i=1}^k \subseteq \mathbb{R}^k$, Δ, $\boldsymbol{c}$;
Output: Δ-separated gridless approximation $\{\hat{\boldsymbol{\mu}}_i\}_{i=1}^k$;

```
{μ̂_i}_{i=1}^k ← {μ_i}_{i=1}^k
for i = 1 → k − 1 do
    l ← arg min_{l⩾i} ||μ̂_l − c||_2, swap(μ̂_i, μ̂_l)
    for j = i + 1 → k do
        if ||μ̂_i − μ̂_j||_2 < Δ then
            λ ← ((μ̂_i − c)^T (μ̂_j − c) + Δ||μ̂_j − c||_2) / ||μ̂_j − c||_2^2
            μ̂_j ← μ̂'_j ≜ c + λ(μ̂_j − c)
        end if
    end for
end for
return {μ̂_i}_{i=1}^k
```

$$\lambda \leftarrow \frac{(\hat{\boldsymbol{\mu}}_i-\boldsymbol{c})^T(\hat{\boldsymbol{\mu}}_j-\boldsymbol{c})+\Delta||\hat{\boldsymbol{\mu}}_j-\boldsymbol{c}||_2}{||\hat{\boldsymbol{\mu}}_j-\boldsymbol{c}||_2^2}$$

$$\hat{\boldsymbol{\mu}}_j \leftarrow \hat{\boldsymbol{\mu}}'_j \triangleq \boldsymbol{c} + \lambda(\hat{\boldsymbol{\mu}}_j - \boldsymbol{c})$$



(a) (b)

Fig. 1: (a) An instance of Algorithm 5 with $k = 5$ and $\Delta = 1$. All $\{\boldsymbol{\mu}_i\}_{i=1}^k$ are moved away from $\boldsymbol{c}$ iteratively satisfying the Δ-separation constraint. (b) A sample move of $\hat{\boldsymbol{\mu}}_j$ to $\hat{\boldsymbol{\mu}}'_j$ with respect to $\hat{\boldsymbol{\mu}}_i, i < j$.

tight. □

3. REFERENCES

[1] S. A. Vavasis, "Complexity theory: quadratic programming", *Encyclopedia of optimization*, Springer, Boston, 2001, pp. 304–307.