Kernelized Few-shot Object Detection with Efficient Integral Aggregation (Supplementary Material)

Shan Zhang[*,†], Lei Wang[♦], Naila Murray[♣], Piotr Koniusz[*,§,†]
[†]Australian National University [♦]University of Wollongong [§]Data61/CSIRO [♣]Meta AI Research

[†]firstname.lastname@anu.edu.au, [♦]leiw@uow.edu.au, [♣]murrayn@fb.com

A. Details of Encoding Network.

Let $f^{\downarrow}:\mathbb{R}^{W\times H}\times\mathbb{R}^{|\mathcal{F}|}\rightarrow\mathbb{R}^{K\times N}$ be the output (of layer 4 of ResNet-50) of Encoder Network in Figure 2a. W and H are the width and height of an input, $N = N_W \cdot N_H$ and K are the total number of spatial locations and features (channel size) in the feature map, respectively. $\mathcal{F}$ are network parameters. Support and query maps from layer 4 are denoted by $\mathbf{\Phi}^{\downarrow}\in\mathbb{R}^{K\times N}$ and $\mathbf{\Phi}^{\downarrow *}\in\mathbb{R}^{K\times N^*}$, where $\mathbf{\Phi}^{\downarrow}=f^{\downarrow}(\boldsymbol{X};\mathcal{F})$ and $\mathbf{\Phi}^{\downarrow *}=f^{\downarrow}(\boldsymbol{X}^*;\mathcal{F})$. The support crop and the query image are denoted as $\mathbf{X}\in\mathbb{R}^{W\times H}$ and $\mathbf{X}^*\in\mathbb{R}^{W^*\times H^*}$. Query features from layer 4 are used by ARPN. Moreover, let $\mathbf{\Phi}=\text{Upscale}(\mathbf{\Phi}^{\downarrow})+f(\boldsymbol{X};\mathcal{F})$ and $\mathbf{\Phi}^*=\text{Upscale}(\mathbf{\Phi}^{\downarrow *})+f(\boldsymbol{X}^*;\mathcal{F})$, where $\mathbf{\Phi}\in\mathbb{R}^{K\times 4N}$ and $\mathbf{\Phi}^*\in\mathbb{R}^{K\times 4N^*}$, f is the output of layer 3 passed via 1×1 convolution ($512\rightarrow 1024$ channel size) combined with the upsampled (by 2×) output $f^{\downarrow}$ of layer 4. Such mixed feature maps (support and query) are used for formation of RKHS matrices and k-autocorrelations (via IRA and/or KA units) passed to ARPN (support representaions only) and MRN (both query and support representations).

B. Full KFSOD Pipeline

Figure 8 illustrates our full pipeline. In the main paper the individual blocks correspond to the Encoding Network (Fig. 2a), the Kernelized Block (Fig. 3a) with the KA unit (Fig. 4) and the Multi-head Relation Net (Fig. 2b).

C. Count sketching as Feature-level Augmentation

1. To see that $\left\langle\frac{K}{K'}\mathbf{P}^{\dagger}\mathbf{P}\boldsymbol{\phi},\boldsymbol{\phi}'\right\rangle$ is a point-wise noisy convolution with variance $\sigma^{\dagger 2}=\frac{1}{K'}(\left\langle\boldsymbol{\phi},\boldsymbol{\phi}'\right\rangle+1)\leq\frac{2}{K'}$, we notice

$$\left\langle\mathbf{P}\boldsymbol{\phi},\mathbf{P}\boldsymbol{\phi}'\right\rangle=\boldsymbol{\phi}^T\mathbf{P}^T\mathbf{P}\boldsymbol{\phi}'=\left\langle\mathbf{P}^T\mathbf{P}\boldsymbol{\phi},\boldsymbol{\phi}'\right\rangle.$$

We also notice that the inverse of the sketching projection is given by $\mathbf{P}^{\dagger}=\frac{K'}{K}\mathbf{P}^T$ because $\mathbf{P}$ contains the orthonormal basis by design (which is scaled by $\frac{K}{K'}$).

Given that §3 tells us that the mean of $\left\langle\mathbf{P}\boldsymbol{\phi},\mathbf{P}\boldsymbol{\phi}'\right\rangle$ over valid projection choices $\mathbf{P}$ is convergent to $\left\langle\boldsymbol{\phi},\boldsymbol{\phi}'\right\rangle$, and the variance bounded by $\frac{1}{K'}(\left\langle\boldsymbol{\phi},\boldsymbol{\phi}'\right\rangle^2+\|\boldsymbol{\phi}\|_2^2\|\boldsymbol{\phi}'\|_2^2)$, it readily follows that if $\|\boldsymbol{\phi}\|_2^2=\|\boldsymbol{\phi}'\|_2^2)=1$ then

$$\left\langle\tfrac{K}{K'}\mathbf{P}^{\dagger}\mathbf{P}\boldsymbol{\phi},\boldsymbol{\phi}'\right\rangle=\left\langle\mathbf{P}\boldsymbol{\phi},\mathbf{P}\boldsymbol{\phi}'\right\rangle\sim\mathcal{N}(\left\langle\boldsymbol{\phi},\boldsymbol{\phi}'\right\rangle,\sigma^{\dagger 2}).$$

2. For injecting the Gaussian noise [46], we have $\left\langle\boldsymbol{\phi}+\Delta\boldsymbol{\phi},\boldsymbol{\phi}'\right\rangle\sim\mathcal{N}(\left\langle\boldsymbol{\phi},\boldsymbol{\phi}'\right\rangle,\sigma^{\ddagger 2})$ for $\left\langle\boldsymbol{\phi}+\Delta\boldsymbol{\phi},\boldsymbol{\phi}'\right\rangle$ where $\Delta\boldsymbol{\phi}\sim\mathcal{N}(\mathbf{0},\sigma^{\ddagger 2})$ because

$$\left\langle\boldsymbol{\phi}+\Delta\boldsymbol{\phi},\boldsymbol{\phi}'\right\rangle=\left\langle\boldsymbol{\phi},\boldsymbol{\phi}'\right\rangle+\left\langle\Delta\boldsymbol{\phi},\boldsymbol{\phi}'\right\rangle$$

and each individual coefficient $\Delta\phi_i\sim\mathcal{N}(0,\sigma^{\ddagger 2})$ so

$$\Big(\sum_i\phi'_i\Delta\phi_i\Big)\sim\mathcal{N}\Big(0,(\phi'^2_1+\cdots+\phi'^2_K)\sigma^{\ddagger 2}\Big)$$

and including $\left\langle\boldsymbol{\phi},\boldsymbol{\phi}'\right\rangle$ into the above can be achieved as

$$\Big(\sum_i\frac{1}{K}\left\langle\boldsymbol{\phi},\boldsymbol{\phi}'\right\rangle+\phi'_i\Delta\phi_i\Big)$$
$$\sim\mathcal{N}\Big(K\frac{1}{K}\left\langle\boldsymbol{\phi},\boldsymbol{\phi}'\right\rangle,(\phi'^2_1+\cdots+\phi'^2_K)\sigma^{\ddagger 2}\Big).\quad(15)$$

Now let $\|\boldsymbol{\phi}\|_2^2=\|\boldsymbol{\phi}'\|_2^2)=1$, we readily obtain

$$\left\langle\boldsymbol{\phi}+\Delta\boldsymbol{\phi},\boldsymbol{\phi}'\right\rangle\sim\mathcal{N}(\left\langle\boldsymbol{\phi},\boldsymbol{\phi}'\right\rangle,\sigma^{\ddagger 2}).$$

The above two derivations exploit the well-known rules for operating on random variables that are independent and normally distributed:

- if $x\sim\mathcal{N}(\mu_1,\sigma_1^2)$ and $y\sim\mathcal{N}(\mu_2,\sigma_2^2)$ then $x+y\sim\mathcal{N}(\mu_1+\mu_2,\sigma_1^2+\sigma_2^2)$,
- if $x\sim\mathcal{N}(\mu,\sigma^2)$ and $k>0$ is a constant then $k\cdot x\sim\mathcal{N}(k\mu,k^2\sigma^2)$.

The above two derivations complete the proof.

[*] Equal contribution. PK is the corresponding author.
Code: https://github.com/ZS123-lang/KFSOD.

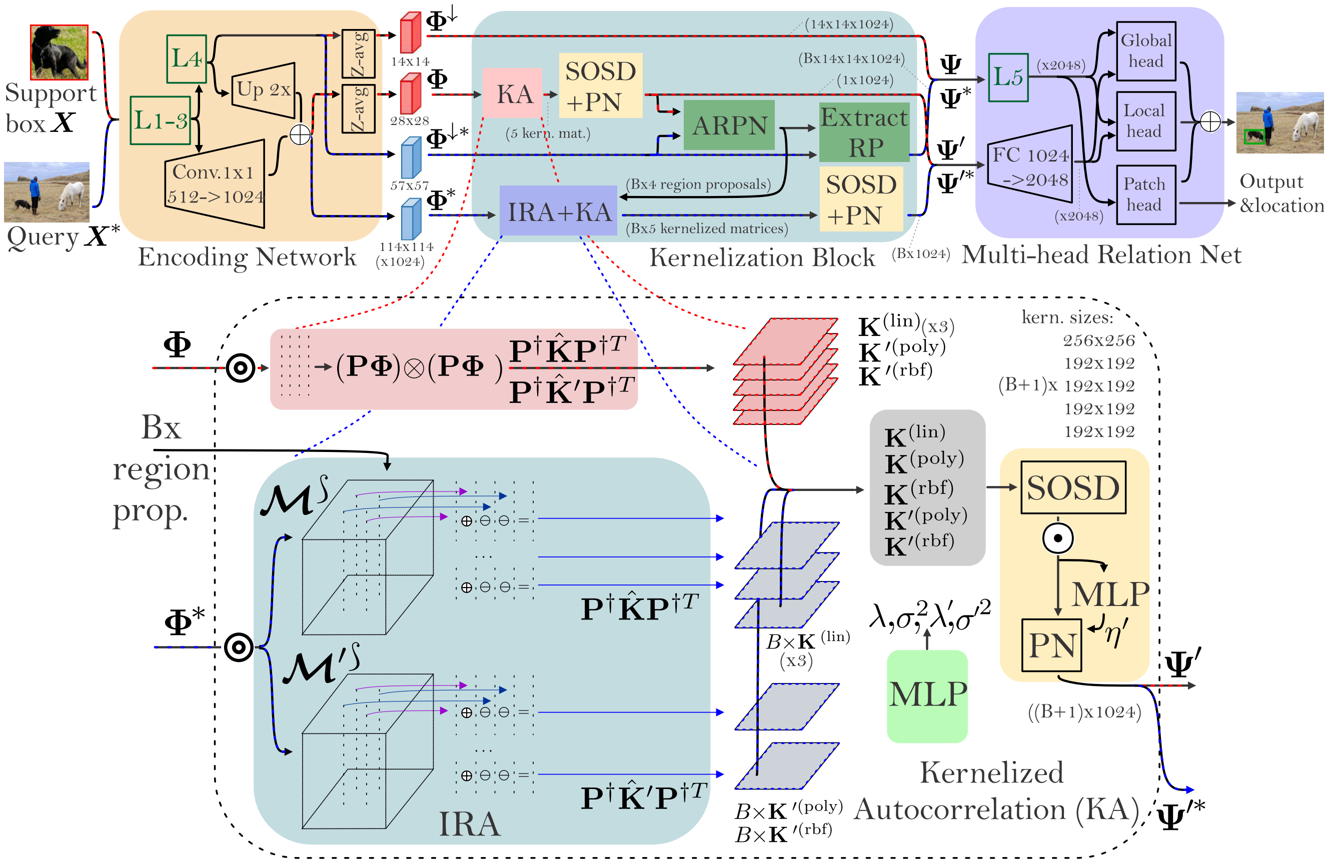

Support box $\boldsymbol{X}$
Query $\boldsymbol{X}^*$
L4
L1-3
Up 2x
Conv.1x1 512->1024
Z-avg
Z-avg
Encoding Network
$\Phi^{\downarrow}$
14x14
Φ
28x28
$\Phi^{\downarrow *}$
57x57
Φ^*
114x114 (x1024)
KA
(5 kern. mat.)
SOSD +PN
ARPN
Extract RP
SOSD +PN
IRA+KA
(Bx4 region proposals)
(Bx5 kernelized matrices)
(14x14x1024)
(Bx14x14x1024)
(1x1024)
(Bx1024)
Kernelization Block
Ψ
Ψ^*
Ψ'
Ψ'^*
L5
(x2048)
FC 1024 ->2048
(x2048)
Global head
Local head
Patch head
Output &location
Multi-head Relation Net
Φ
Bx region prop.
Φ^*
$(\mathbf{P}\Phi)\otimes(\mathbf{P}\Phi)$
$\mathbf{P}^{\dagger}\hat{\mathbf{K}}\mathbf{P}^{\dagger T}$
$\mathbf{P}^{\dagger}\hat{\mathbf{K}}'\mathbf{P}^{\dagger T}$
$\mathcal{M}^{\int}$
$\mathcal{M}'^{\int}$
$\mathbf{P}^{\dagger}\hat{\mathbf{K}}\mathbf{P}^{\dagger T}$
$\mathbf{P}^{\dagger}\hat{\mathbf{K}}'\mathbf{P}^{\dagger T}$
IRA
$\mathbf{K}^{(\text{lin})}$(x3)
$\mathbf{K}'^{(\text{poly})}$
$\mathbf{K}'^{(\text{rbf})}$
kern. sizes:
256x256
192x192
(B+1)x 192x192
192x192
192x192
$\mathbf{K}^{(\text{lin})}$
$\mathbf{K}^{(\text{poly})}$
$\mathbf{K}^{(\text{rbf})}$
$\mathbf{K}'^{(\text{poly})}$
$\mathbf{K}'^{(\text{rbf})}$
SOSD
MLP
PN
η'
$\lambda,\sigma^2,\lambda',\sigma'^2$
MLP
$B\times\mathbf{K}^{(\text{lin})}$ (x3)
$B\times\mathbf{K}'^{(\text{poly})}$
$B\times\mathbf{K}'^{(\text{rbf})}$
Ψ'
((B+1)x1024)
Ψ'^*
Kernelized Autocorrelation (KA)

Figure 8. The full KFSOD pipeline includes the Encoding Network, the Kernelization Block, and the Multi-head Relation Network. The Kernelization Autocorrelation (KA) unit includes IRA to process the large number of region proposals in the query image. Note that we have also indicated the inverse count sketching step in KA for better clarity.

D. Details of Multi-head Relation Net

Figure 9a is the Multi-head Relation Network that contains the Global head (Fig. 9b), the Local head (Fig. 9c) and the Patch head (Fig. 9c). The role of each head is described in the caption below Figure 9. The individual blocks and their parameters are also included.

E. Class-wise performance on PASCAL VOC 2007

Table 9 shows the performance of our KFSOD on novel and base classes. The table shows that on average, we achieve almost 8.4% boost over PSND given the novel classes protocol, and 4.1% boost over PSND given the base classes protocol.

F. Performance of Group of Kernels

Table 10 demonstrates that it is beneficial to use numerous kernels in our pipeline. Firstly, we note that all kernelized representations were equipped with MLP to learn their hyper-parameters. Moreover, the kernel pooling step uses MLP and default parameters (as in the main paper). Any other parameters were selected by the crossvalidation on the validation split of the PASCAL VOC 2007.

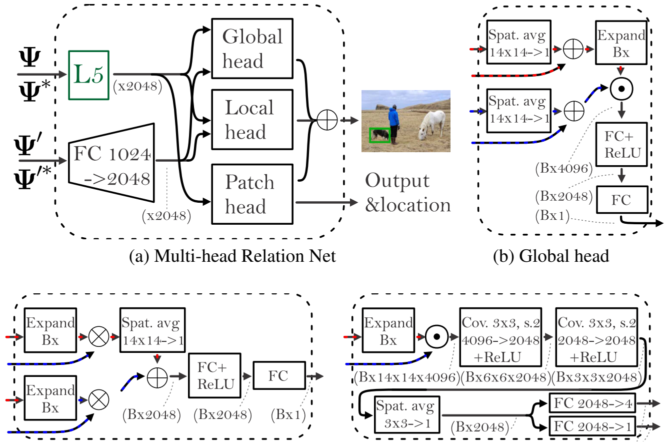
Firstly, we notice that the single RKHS linear kernel, denoted as Lin/k, is the worst performer despite utilizing 1024 dimensions of the feature map channel mode. Using four linear kernels (Lin/k) is marginally better and six linear kernels (Lin/k) are even better. Despite this may feel unexpected, in fact, six linear kernels enjoy 6 MLP units in the kernel pooling step, each likely yielding somewhat different η' which makes each of these six kernels act differently.

Moreover, using two kernelized representations RBF/k and RBF/a, denoted as RBF/k+a, each enjoying separate 512 dimensions of the feature map channel mode, appears to be better that any number of linear kernels combined. Moreover, polynomial representations appear slightly worse than RBF representations.

Combining four kernelized representations, that is 2× RBF/k and 2× RBF/a improves resutls further. However, combining four distinct kernelized representations, that is

Table 9. Comparison with SOTA on the PASCAL VOC 2007 testing set (class split 1, 5-shot protocol) in terms of mAP %.

Method		Novel bird	bus	cow	mbike	sofa	mean ± std	Base aero	bike	boat	bottle	car	cat	chair	table	dog	horse	person	plant	sheep	train	tv	mean ± std
FR	ICCV19	20.0	48.7	29.2	47.9	24.6	33.9±12.0	65.3	73.5	54.7	39.5	75.7	81.1	35.3	62.5	72.8	78.8	68.6	41.5	59.2	76.2	69.2	63.6±14.3
FRCN	ICCV12	31.3	36.9	54.1	26.5	36.2	36.9±9.3	68.4	75.2	59.2	54.8	74.1	80.8	42.8	56.0	68.9	77.8	75.5	34.7	66.1	71.2	66.2	64.8±12.9
LSTD	AAAI18	22.8	52.5	31.3	45.6	40.3	38.5±10.5	70.9	71.3	59.8	41.1	77.1	81.9	45.1	67.2	78.0	78.9	70.7	**41.6**	63.8	79.7	66.8	66.3±13.3
Meta	ICCV19	48.5	49.9	49.7	48.6	41.6	45.7±3.7	68.1	73.9	59.8	54.2	80.1	82.9	48.8	62.8	80.1	81.4	77.2	37.2	65.7	75.8	70.6	67.9±12.9
NP-RepMet	NeurIPS20	16.8	62.1	49.1	55.8	52.7	47.3±15.8	71.9	**79.1**	64.9	**70.8**	73.6	49.5	**53.5**	67.3	62.7	78.7	74.8	58.3	76.2	72.5	67.9	68.3±8.6
FSOD	CVPR20	56.5	57.9	53.7	56.6	52.8	55.5±5.2	67.0	72.3	57.6	53.1	78.5	80.7	47.4	61.9	78.1	82.6	75.3	35.6	64.2	74.4	69.1	66.1±13.0
PNSD	ACCV20	57.6	60.2	54.3	55.6	54.8	56.5±5.9	69.3	74.8	61.5	53.4	80.2	82.3	49.6	61.8	80.8	82.6	77.9	35.8	68.6	78.2	**70.9**	68.5±13.3
KFSOD	(Ours)	**60.5**	**66.7**	**59.6**	**62.3**	**55.4**	**60.9±3.7**	**72.5**	78.1	**66.7**	68.3	**81.0**	**85.1**	52.2	**67.7**	**83.3**	**84.0**	**82.5**	41.3	**76.6**	**82.5**	70.5	**72.6±12.1**



(a) Multi-head Relation Net

(b) Global head

(c) Local head

(d) Patch head

Figure 9. Multi-head Relation Net (Fig. 9a) receives first-order representations ($\mathbf{\Psi}$ for support, $\mathbf{\Psi}^*$ for the query) and kernelized representations ($\mathbf{\Psi}'$ for support, $\mathbf{\Psi}'^*$ for the query) from the Kernelization Block. From first-order maps, layer L5 of ResNet-50 generates feature maps with 2048 channels. Similarly, an FC layer maps representations of 1024 to 2048 dimensional space. Such feature maps are fed into the global, local and patch heads in Fig. 9b, 9c and 9d. Operators ⊕, ⊙ and ⊗ are addition, channel-wise concatenation and element-wise multiplication. Block (Expand Bx) replicates the support feature map B times to match its size with the query map containing descriptors of B candidate regions.

kernels	size	Kernel Lin/k	RBF/k+a	Poly/k+a	Pair	All	Shot/5 Novel	Base
one	1024	✓					50.8	64.8
two	512×2		✓				52.5	66.3
two	512×2			✓			52.2	66.0
four	256×4	✓					51.4	65.2
four	256×4		✓				52.9	67.3
four	256×4			✓			53.1	67.5
four	256×4				✓		54.9	69.2
six	(128+192+192)×2	✓					51.8	65.7
six	(128+192+192)×2		✓				53.3	68.1
six	(128+192+192)×2			✓			53.8	68.0
five	256+192×4					✓	**57.1**	**70.3**

Table 10. Ablations (mAP) on PASCAL VOC 2007 (5-shot, novel and base classes) w.r.t. the different number of RKHS kernels and k-autocorrelations combined. As we split feature maps along the channel mode by operator ⊙ to obtain several groups, each per one kernelized representation, we indicate how many channel dimensions are used by the column (*size*). Note that all kernelized representations used in this ablation study are equipped with MLP (except for Lin/k that has no hyper-parameter). By (*Pair*) we denote four kernelized representations (RBF/k+a)+(Poly/k+a). By (*All*) we denote five kernelized representations (Lin/k)+(RBF/k+a)+(Poly/k+a).

Resnet-50 5-shot (VOC)	5-shot (FSOD)	10-shot (COCO)	Swin-B7 5-shot (VOC)	5-shot (FSOD)	10-shot (COCO)	Swin-B12 5-shot (VOC)	5-shot (FSOD)	10-shot (COCO)
60.9	31.7	22.9	**62.1**	**32.6**	**24.1**	61.2	31.8	22.5

Table 11. Ablations w.r.t. different EN backbones on PASCAL VOC 2007, FSOD, and COCO dataset (5/10-shot, novel classes).

RBF/k, RBF/a, Poly/k and Poly/a is much better than utilizing the same kernelized representation twice.

Finally, combining five distinct kernelized representations, that is Lin/k, RBF/k, RBF/a, Poly/k and Poly/a resulted in the best performance. **Thus, we use this combination of kernelized representations on all datasets.**

G. Ablation Study on Encoding Network

Below we perform ablations of the backbone (Encoding Network, termed as EN in main paper). We use ConvNet (ResNet-50) and Transformer network [26] (Swin-B^7/ Swin-B^{12} pre-trained on ImageNet-22K [4] with window size of 7/12), as shown in Table 11. The comparisons are conducted by changing the backbone, whereas other settings remain unchanged. When ResNet-50 is replaced by Swin-B^7, we gain an improvement of ∼1.0% on all three dataset, in the 5/10-shot setting (novel classes).

H. Implementation details

KFSOD uses ResNet-50 pre-trained on ImageNet [4] and MS COCO [24]. We fine-tune the network with a learning rate of 0.002 for the first 56000 iterations and 0.0002 for another 4000 iterations. Images are resized to 600 pixels (shorter edge) and the longer edge is capped at 1000 pixels. Each support image is cropped based on ground-truth boxes, bilinearly interpolated and padded to 320×320.

I. Hyper-parameters used in experiments

- **PASCAL VOC 2007:**
  RBF/k+a ($\sigma=\sigma'=0.1$ only in ablations where MLP is indicated as not used, otherwise $\kappa=\kappa'=0.2$ is set for MLP);
  Poly/k+a $r=r'=10$, ($\lambda=\lambda'=1$ only in ablations where MLP is indicated as not used, otherwise $\kappa=\kappa'=1$ is set for MLP);
  kernel pooling $\eta=9$ and $\kappa''=700$.

- **FSOD:**
  RBF/k+a ($\sigma=\sigma'=1$ only in ablations where MLP is indicated as not used, otherwise $\kappa=\kappa'=1.3$ is set for MLP);
  Poly/k+a $r=r'=3$, ($\lambda=\lambda'=1$ only in ablations where MLP is indicated as not used, otherwise $\kappa=\kappa'=4$ is set for MLP);
  kernel pooling $\eta=5$ and $\kappa''=500$.

- **COCO:**
  RBF/k+a ($\sigma=\sigma'=0.2$ only in ablations where MLP is indicated as not used, otherwise $\kappa=\kappa'=0.4$ is set for MLP);
  Poly/k+a $r=r'=7$, ($\lambda=\lambda'=1$ only in ablations where MLP is indicated as not used, otherwise $\kappa=\kappa'=1$ is set for MLP);
  kernel pooling $\eta=5$ and $\kappa''=500$.