Supplementary material for CVPR 2019 paper #6535

Yeonwoo Jeong, Yoonsung Kim, Hyun Oh Song
Department of Computer Science and Engineering, Seoul National University, Seoul, Korea
{yeonwoo, yskim227, hyunoh}@mllab.snu.ac.kr

A. Proofs for equivalence

A.1 Recap for flow definition

$$
\begin{aligned}
&(i)\ f_z(s, a_p) = k_s\\
&(ii)\ f_z(a_p, b_{r,q}) = \mathbf{z}_p[q]\\
&(iii)\ f_z\left((b_{r,q}, b_{0,q})_i\right) = \begin{cases} 1 & \forall i < \sum_{p:a_p\in A_r} \mathbf{z}_p[q] \\ 0 & \text{otherwise} \end{cases}\\
&(iv)\ f_z\left((b_{0,q}, t)_j\right) = \begin{cases} 1 & \forall j < \sum_{p=1}^{n_c} \mathbf{z}_p[q] \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
\tag{1}
$$

A.2 Proofs for lemma1 and lemma2

Lemma 1. *Given the minimum cost flow $\{f_o(e)\}_{e\in E'}$ of the network G', the total cost of the flow is $\sum_{e\in E'} v(e)f_o(e) = \sum_{p=1}^{n_c} -\mathbf{c}_p{}^T\mathbf{z}'_p + \sum_{r=1}^{l}\sum_{p_1\neq p_2\in\{p|a_p\in A_r\}} \alpha\mathbf{z}'_{p_1}{}^T\mathbf{z}'_{p_2} + \sum_{p_1\neq p_2}\beta\mathbf{z}'_{p_1}{}^T\mathbf{z}'_{p_2}$.*

Proof. The total minimum cost flow is

$$
\begin{aligned}
\sum_{e\in E'} v(e)f_o(e) = &\underbrace{\sum_{a_p\in A} v(s,a_p)f_o(s,a_p)}_{\text{Flow from source to vertices in } A} +\\
&\underbrace{\sum_{r=0}^{l}\sum_{a_p\in A_r}\sum_{b_{r,q}\in B_r} v(a_p,b_{r,q})f_o(a_p,b_{r,q})}_{\text{Flow from vertices in } A \text{ to vertices in } B_r} +\\
&\underbrace{\sum_{r=1}^{l}\sum_{b_{r,q}\in B_r}\sum_{i=0}^{g_r-1} v((b_{r,q},b_{0,q})_i)f_o((b_{r,q},b_{0,q})_i)}_{\text{Flow from vertices in } B_r \text{ to vertices in } B_0} +\\
&\underbrace{\sum_{b_{0,q}\in B_0}\sum_{j=0}^{n_c-1} v((b_{0,q},t)_j)f_o((b_{0,q},t)_j)}_{\text{Flow from vertices in } B_0 \text{ to sink}}
\end{aligned}
$$

Also, for $r > 0$, denote the amount of input flow at each vertex $b_{r,q} \in B_r$ given the minimum cost flow as $y'_{r,q} = \sum_{a_p\in A_r} f_o(a_p, b_{r,q}) = \sum_{p:a_p\in A_r} \mathbf{z}'_p[q]$. Also, denote the amount of input flow at each vertex $b_{0,q}\in B_0$ as $y'_{0,q} = \sum_{p:a_p\in A_0} f_o(a_p, b_{0,q}) + \sum_{r=1}^{l} y'_{r,q} = \sum_{p=1}^{n_c}\mathbf{z}'_p[q]$. Then, from the optimality of the minimum cost flow, $f_o((b_{r,q}, b_{0,q})_i) = \begin{cases} 1 & \forall\, i < y'_{r,q} \\ 0 & \text{otherwise}\end{cases}$ and $f_o((b_{0,q},t)_j) = \begin{cases} 1 & \forall j < y'_{0,q} \\ 0 & \text{otherwise}\end{cases}$. Therefore, the total cost for optimal flow is

$$
\begin{aligned}
&\sum_{e\in E'} v(e)f_o(e) = 0 + \sum_{r=0}^{l}\sum_{a_p\in A_r}\sum_{b_{r,q}\in B_r} -\mathbf{c}_p[q]\mathbf{z}'_p[q] +\\
&\quad\sum_{r=1}^{l}\sum_{b_{r,q}\in B_r}\sum_{i=0}^{y'_{r,q}-1} 2\alpha i + \sum_{b_{0,q}\in B_0}\sum_{j=0}^{y'_{0,q}-1} 2\beta j\\
&= \sum_p -\mathbf{c}_p^T\mathbf{z}'_p + \sum_{r=1}^{l}\sum_{b_{r,q}\in B_r}\alpha y'_{r,q}(y'_{r,q}-1) + \sum_{b_{0,q}\in B_0}\beta y'_{0,q}(y'_{0,q}-1)\\
&= \sum_p -\mathbf{c}_p^T\mathbf{z}'_p + \sum_{r=1}^{l}\sum_{b_{r,q}\in B_r}\alpha {y'_{r,q}}^2 - \sum_{r=1}^{l}\sum_{p:a_p\in A_r}\sum_{q=1}^{d}\alpha\mathbf{z}'_p[q] +\\
&\quad\sum_{b_{0,q}\in B_0}\beta {y'_{0,q}}^2 - \sum_{p=1}^{n_c}\sum_{q=1}^{d}\beta\mathbf{z}'_p[q]\\
&= \sum_p -\mathbf{c}_p^T\mathbf{z}'_p + \alpha\sum_{r=1}^{l}\sum_{p:a_p\in A_r}{\mathbf{z}'_p}^T\sum_{p:a_p\in A_r}\mathbf{z}'_p - \alpha\sum_{r=1}^{l}\sum_{p:a_p\in A_r}\mathbf{z}'_p{}^T\mathbf{z}'_p\\
&\quad+\beta\sum_{p=1}^{n_c}{\mathbf{z}'_p}^T\sum_{p=1}^{n_c}\mathbf{z}'_p - \beta\sum_{p=1}^{n_c}\mathbf{z}'_p{}^T\mathbf{z}'_p\\
&= \sum_{p=1}^{n_c} -\mathbf{c}_p{}^T\mathbf{z}'_p + \sum_{r=1}^{l}\sum_{p_1\neq p_2\in\{p|a_p\in A_r\}}\alpha\mathbf{z}'_{p_1}{}^T\mathbf{z}'_{p_2} + \sum_{p_1\neq p_2}\beta\mathbf{z}'_{p_1}{}^T\mathbf{z}'_{p_2}.
\end{aligned}
$$

□

Lemma 2. *Given a feasible flow $\{f_z(e)\}_{e\in E'}$ of the network G', the total cost of the flow is $\sum_{e\in E'} v(e)f_z(e) = \sum_{p=1}^{n_c} -\mathbf{c}_p{}^T\mathbf{z}_p + \sum_{r=1}^{l}\sum_{p_1\neq p_2\in\{p|a_p\in A_r\}}\alpha\mathbf{z}_{p_1}{}^T\mathbf{z}_{p_2} + \sum_{p_1\neq p_2}\beta\mathbf{z}_{p_1}{}^T\mathbf{z}_{p_2}$.*

Proof. The total cost proof is similar to Lemma 1 except that we use the flow conditions from Equation (1) (iii) and

Equation (1) (iv) instead of the optimality of the flow.

$$\sum_{e\in E'} v(e)f_z(e) = 0 + \sum_{r=0}^{l}\sum_{a_p\in A_r}\sum_{b_{r,q}\in B_r} -\mathbf{c}_p[q]\mathbf{z}_p[q] +$$

$$\sum_{r=1}^{l}\sum_{b_{r,q}\in B_r}\sum_{i=0}^{y_{r,q}-1} 2\alpha i + \sum_{b_{0,q}\in B_0}\sum_{j=0}^{y_{0,q}-1} 2\beta j$$

$$= \sum_{p} -\mathbf{c}_p^T\mathbf{z}_p + \sum_{r=1}^{l}\sum_{b_{r,q}\in B_r} \alpha y_{r,q}(y_{r,q}-1) + \sum_{b_{0,q}\in B_0} \beta y_{0,q}(y_{0,q}-1)$$

$$= \sum_{p} -\mathbf{c}_p^T\mathbf{z}_p + \sum_{r=1}^{l}\sum_{b_{r,q}\in B_r} \alpha {y_{r,q}}^2 - \sum_{r=1}^{l}\sum_{p:a_p\in A_r}\sum_{q=1}^{d} \alpha\mathbf{z}_p[q] +$$

$$\sum_{b_{0,q}\in B_0} \beta {y_{0,q}}^2 - \sum_{p=1}^{n_c}\sum_{q=1}^{d} \beta\mathbf{z}_p[q]$$

$$= \sum_{p} -\mathbf{c}_p^T\mathbf{z}_p + \alpha\sum_{r=1}^{l}\sum_{p:a_p\in A_r} {\mathbf{z}_p}^T \sum_{p:a_p\in A_r}\mathbf{z}_p - \alpha\sum_{r=1}^{l}\sum_{p:a_p\in A_r} {\mathbf{z}_p}^T\mathbf{z}_p$$

$$+ \beta{\sum_{p=1}^{n_c}\mathbf{z}_p}^T \sum_{p=1}^{n_c}\mathbf{z}_p - \beta\sum_{p=1}^{n_c}{\mathbf{z}_p}^T\mathbf{z}_p$$

$$= \sum_{p=1}^{n_c} -{\mathbf{c}_p}^T\mathbf{z}_p + \sum_{r=1}^{l}\sum_{p_1\neq p_2\in\{p|a_p\in A_r\}} \alpha{\mathbf{z}_{p_1}}^T\mathbf{z}_{p_2} + \sum_{p_1\neq p_2}\beta{\mathbf{z}_{p_1}}^T\mathbf{z}_{p_2}$$

□

B. Time complexity

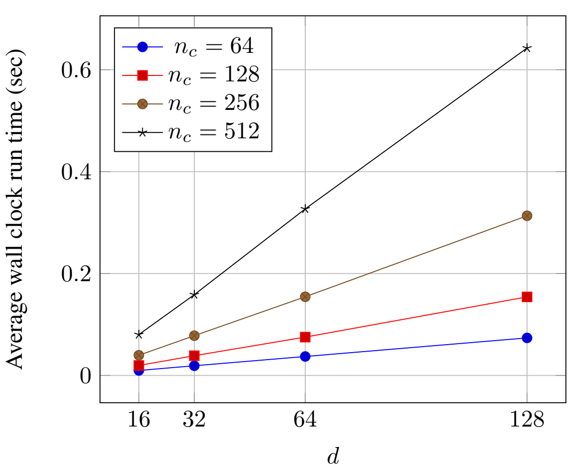

Figure 1: Average wall clock run time of computing minimum cost flow on G' per mini-batch using [2]. In practice, the run time is approximately linear in n_c and d. Each data point is averaged over 20 runs on machines with Intel Xeon E5-2650 CPU.

C. Effect of label remapping

We performed the ablation study without the label remapping method. 'Ours-r' in Table 1 and Table 2 shows the results without the remapping. The ablation study shows that the proposed hierarchical structure leads to much larger improvements than the remapping method.

		test				*train*			
	Method	SUF	Pr@1	Pr@4	Pr@16	SUF	Pr@1	Pr@4	Pr@16
k_s	Metric	1.00	56.78	55.99	53.95	1.00	62.64	61.91	61.22
1	LSH	**138.83**	52.52	48.67	39.71	**135.64**	60.45	58.10	54.00
	Th	41.21	54.82	52.88	48.03	43.19	61.56	60.24	58.23
	VQ	22.78	56.74	55.94	53.77	40.35	62.54	61.78	60.98
	[1]	97.67	57.63	57.16	55.76	97.77	63.85	63.40	63.39
	Ours-r	98.12	58.16	57.39	56.01	98.45	64.38	63.94	63.86
	Ours	97.67	**58.42**	**57.88**	**56.58**	97.28	**64.73**	**64.63**	**64.69**
2	Th	14.82	56.55	55.62	52.90	15.34	62.41	61.68	60.89
	VQ	5.63	56.78	56.00	53.99	6.94	62.66	61.92	61.26
	[1]	76.12	57.30	56.70	55.19	78.28	63.60	63.19	63.09
	Ours-r	**101.26**	58.18	**57.68**	**56.43**	**100.29**	**65.00**	**64.52**	**64.42**
	Ours	98.38	**58.39**	57.51	56.09	97.20	64.35	63.91	63.81
3	Th	7.84	56.78	55.91	53.64	8.04	62.66	61.88	61.16
	VQ	2.83	56.78	55.99	53.95	2.96	62.62	61.92	61.22
	[1]	42.12	56.97	56.25	54.40	44.36	62.87	62.22	61.84
	Ours-r	**97.78**	57.58	57.14	55.70	**97.25**	**63.95**	**63.58**	**63.48**
	Ours	94.55	**58.19**	**57.42**	**56.02**	93.69	63.60	63.35	63.32
4	Th	4.90	56.84	56.01	53.86	5.00	62.66	61.94	61.24
	VQ	1.91	56.77	55.99	53.94	1.97	62.62	61.91	61.22
	[1]	16.19	57.11	56.21	54.20	16.52	62.81	62.14	61.58
	Ours-r	**98.36**	57.58	57.18	55.94	**97.88**	63.90	63.32	63.25
	Ours	92.18	**58.52**	**57.79**	**56.22**	91.27	**64.20**	**63.95**	**63.63**

Table 1: Results with Triplet network with hard negative mining. Querying test data against a hash table built on *test* set and a hash table built on *train* set on Cifar-100.

		test				*train*			
	Method	SUF	Pr@1	Pr@4	Pr@16	SUF	Pr@1	Pr@4	Pr@16
k_s	Metric	1.00	57.05	55.70	53.91	1.00	61.78	60.63	59.73
1	LSH	29.74	53.55	50.75	43.03	30.75	59.87	58.34	55.35
	Th	12.72	54.95	52.60	47.16	13.65	60.80	59.49	57.27
	VQ	34.86	56.76	55.35	53.75	31.35	61.22	60.24	59.34
	[1]	54.85	58.19	57.22	55.87	54.90	**63.11**	62.29	61.94
	Ours-r	95.30	58.04	57.31	56.22	90.63	62.55	62.15	61.77
	Ours	**101.1**	**58.28**	**57.79**	**56.92**	**97.47**	63.06	**62.62**	**62.44**
2	Th	5.09	56.52	55.28	53.04	5.36	61.65	60.50	59.50
	VQ	6.08	57.13	55.74	53.90	5.44	61.82	60.56	59.70
	[1]	16.20	57.27	55.98	54.42	16.51	61.98	60.93	60.15
	Ours-r	66.74	57.73	57.01	55.66	67.46	62.76	61.87	61.36
	Ours	**69.48**	**57.60**	**56.98**	**55.82**	**69.91**	**62.19**	**61.71**	**61.27**
3	Th	3.10	56.97	55.56	53.76	3.21	61.75	60.66	59.73
	VQ	2.66	57.01	55.69	53.90	2.36	61.78	60.62	59.73
	[1]	7.25	57.15	55.81	54.10	7.32	61.90	60.80	59.96
	Ours-r	55.83	**57.81**	56.55	55.11	57.11	62.20	**61.50**	60.90
	Ours	**57.09**	57.56	**56.70**	**55.41**	**58.62**	**62.30**	61.44	**60.91**
4	Th	2.25	57.02	55.64	53.88	2.30	61.78	60.66	59.75
	VQ	1.66	57.03	55.70	53.91	1.55	61.78	60.62	59.73
	[1]	4.51	57.15	55.77	54.01	4.52	61.81	60.69	59.77
	Ours-r	48.04	**57.76**	56.70	55.11	49.73	62.12	61.30	60.74
	Ours	**49.43**	57.75	**56.79**	**55.50**	**50.80**	**62.43**	**61.65**	**61.01**

Table 2: Results with Npairs [4] network. Querying test data against a hash table built on *test* set and a hash table built on *train* set on Cifar-100.

D. ImagenetSplit Detail

Table 3 shows the C_{train} and C_{test} explicitly.

train	*test*
tench,electric ray,cock,jay	goldfish,stingray,hen,magpie
common newt,spotted salamander,tree frog,loggerhead	eft,axolotl,tailed frog,leatherback turtle
common iguana,agama,diamondback,trilobite	American chameleon,frilled lizard,sidewinder,centipede
harvestman,quail,African grey,bee eater	scorpion,partridge,macaw,hornbill
jacamar,echidna,flatworm,crayfish	toucan,platypus,nematode,hermit crab
white stork,little blue heron,red-backed sandpiper,Walker hound	black stork,bittern,redshank,English foxhound
Irish wolfhound,whippet,Staffordshire bullterrier,vizsla	borzoi,Italian greyhound,American Staffordshire terrier,German short-haired pointer
English springer,schipperke,malinois,Siberian husky	Welsh springer spaniel,kuvasz,groenendael,malamute
Cardigan,cougar,meerkat,dung beetle	Pembroke,lynx,mongoose,rhinoceros beetle
ant,grasshopper,cockroach,cicada	bee,cricket,mantis,leafhopper
dragonfly,Angora,hippopotamus,siamang	damselfly,wood rabbit,llama,gibbon
indri,African elephant,lesser panda,sturgeon	Madagascar cat,Indian elephant,giant panda,gar
hamper,bicycle-built-for-two,fireboat,cello	shopping basket,mountain bike,gondola,violin
soup bowl,oxygen mask,china cabinet,Polaroid camera	mixing bowl,snorkel,medicine chest,reflex camera
bottlecap,freight car,swab,fur coat	nipple,passenger car,broom,lab coat
shower curtain,computer keyboard,bassoon,cliff dwelling	theater curtain,joystick,oboe,yurt
loudspeaker,wool,harvester,oil filter	microphone,velvet,thresher,strainer
accordion,bookcase,suit,beer glass	harmonica,wardrobe,diaper,goblet
vestment,acoustic guitar,barrow,space heater	academic gown,electric guitar,shopping cart,stove
crash helmet,vase,whiskey jug,cleaver	football helmet,beaker,water jug,letter opener
candle,airship,combination lock,electric locomotive	spotlight,balloon,padlock,steam locomotive
barometer,stethoscope,binoculars,Dutch oven	scale,syringe,projector,rotisserie
frying pan,grand piano,church,swing	wok,upright,mosque,teddy
knee pad,crossword puzzle,kimono,catamaran	apron,jigsaw puzzle,abaya,trimaran
feather boa,cocktail shaker,holster,apiary	stole,saltshaker,scabbard,boathouse
birdhouse,pirate,bobsled,slot	bell cote,wreck,dogsled,vending machine
canoe,crutch,pay-phone,cinema	yawl,flagpole,dial telephone,home theater
parking meter,moving van,barbell,ice lolly	stopwatch,police van,dumbbell,ice cream
broccoli,zucchini,acorn squash,orange	cauliflower,spaghetti squash,butternut squash,lemon
eggnog,alp,lakeside	cup,volcano,seashore

Table 3: Class names in train dataset and test dataset splitted from [3].

References

[1] Y. Jeong and H. O. Song. Efficient end-to-end learning for quantizable representations. In *ICML*, 2018. 2

[2] G. OR-tools. https://developers.google.com/optimization/, 2018. 2

[3] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 3

[4] K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NIPS*, 2016. 2